track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

Problem 1.3 Robust addition

The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

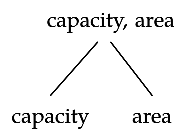
Problem 1.4 Intelligent redundancy

Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

Are nodes and leaves the same thing?

A Node refers to each individual "quantity" that we are finding. A leaf however is a node that does not sprout a new branch. In this tree, 'capacity' is a node while 'sample rate' is a leaf.

lateral movement represents "intelligent" redundancy while longitudinal movement represents divide-and-conquer. I like the trees because they are a more visual way of showing/remembering where numbers came from...though I agree that it would be nice if the longitudinal movements were labeled with equations or something of that sort. I can see these trees becoming quite complicated for other examples though

I thought this section was written pretty clearly. The diagrams make it easy to grasp the concept. Perhaps the wording could be improved. Like someone else said earlier, the first 2 paragraphs are awkwardly worded. Other than that, I have no real comments to add! I hope that's not a bad thing.

I definitely like the visualization of the method. It makes the thought process more clear for people who like pictures.

I don't think this actually helps in estimation. I feel like the difficulty in a lot of what we're doing is figuring out exactly how we're going to estimate something, and I feel like these trees, although new (so it may take some getting used to), maybe just be a wasted way of organizing something that we should be able to handle at any mildly high level of estimation.

I disagree. While many approximations that we've done have been simple enough that this is not necessary, I can imagine more complex ones for which this would be useful. However, much of its use is in looking back on the work you've done when it's no longer fresh in your mind. You can follow your own logic (or explain it to others) very easily with a diagram of this type. It also makes errors easier to find and fix.

track whose 'rings' lie 1.6 µm apart. Along the track, the pits lie 0.9 µm apart. So, the spacing is between 0.9 and 1.6 µm; if you want just one value, let it be the midpoint, 1.3 µm. We made a tasty pudding!

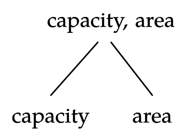
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

The teaching style of this class is so different than in other classes. I think it is interesting that all of the lessons from the reading all use the same example of a CD-ROM. Would it be more beneficial if the text had a different example or used more than one (maybe its just that I don't find CD-ROMs very interesting)?

I agree. But this is only based on the first three readings. I don't expect the readings to continue to be entirely about CD's. Look at example 1.4, which looks much more interesting to me, it has no CD's.

while i realize that, technically, there haven't actually been that many book sections on CDs, CDs have been the major focus for about a week's worth of class. i'm not very comfortable with 8.02, waves, digital information sizes, and electrical engineering, so despite sanjoy's best efforts, i still find CDs very difficult to understand.

i feel like, thus far (at least in the readings), i have been struggling more to understand CDs than to understand approximation methods.

This comment and many others along the same lines – all of which I agree with – have pretty much convinced me to significantly rewrite these first few sections. The main idea is divide and conquer, and it's getting lost in the details in my current organization.

I think the first example will be, as suggested on one of the paper end-of-lecture A5 sheets, estimating the number of seconds in a year. And use a tree from the beginning.

Once divide and conquer is introduced, then maybe do something like the capacity of a CD or the bandwidth of a 747 filled with CDs.

Why is looking at the box a viable option? Can't be deceiving? Also, how does that really teach us anything about the item/measurement? Shouldn't it be an option we use to make other measurements?

track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

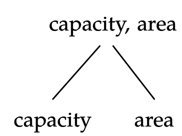
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

It would be really nice to see an example that hadn't already been done in addition, or maybe before, the cd one that we are familiar with. It could be helpful to see both how old ways of thinking about it translate and how the tree can help you think differently.

I agree to a certain extent... the CD example in my opinion is a little too technical for someone to be able to think up off the top of their head... but its nice reiterating a process that was already done for sheer sake of understanding the tree approach. Because of this, it allows the reader to be able to relate to something previously done.

So does the tree just serve to help you break down equations? Usually to make the tree what I looking for and then think of equations I need to find it and place the variables in the equations in branches below.

track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

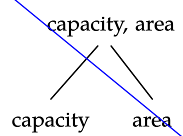
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

Are you going to include solutions to these problems (for readers using the book for self-study, for example)?

Read this section for Wednesday's lecture; the memo due Tuesday at 10pm. I'll have the links from the 6.055 website point to the new NB user interface (the old UI seems to be a bit fragile right now). Let me and Sacha know if you have problems with NB.

Is there a way to get back to the "File" and "Nav" tabs on the left frame and the "Home" tab on the right frame after clicking to see our own comments besides clicking "Back" on the browser? I tried closing the "Controls" and "Notes", which resulted in just white empty frames.

I've forwarded that question to Sacha, who hopefully can say how to do it or will implement an answer!

I don't feel like this section adds much new information. The tree structure seems helpful but doesn't need much explanation. The bigger thing that I think will trip me and others up is not having the numbers at my fingertips with which to do these estimates. I feel like a list of numbers to memorize might be a good start to this class. Even if it's boring, it would help give us a foundation of numbers to work from.

Meh...I disagree to a certain extent. If you just memorize things, you might start thinking too rigidly. It's more useful if you just play around with things and estimate based on your intuition. Eventually, this playing around will allow you to naturally memorize certain values/constants w/o the pain of memorizing them for memorization's sake.

estimation pudding? I have no idea what this is referring to...

It's a metaphor meant to capture your attention and make the reading more entertaining. Or just make you hungry.

It's a metaphor meant to capture your attention and make the reading more entertaining. Or just make you hungry.

The dash here seems a bit awkward, though maybe I'm reading too much into it.

It kinda makes sense to me, like a mental break

track whose 'rings' lie 1.6 µm apart. Along the track, the pits lie 0.9 µm apart. So, the spacing is between 0.9 and 1.6 µm; if you want just one value, let it be the midpoint, 1.3 µm. We made a tasty pudding!

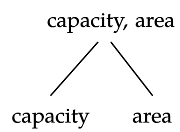
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

This whole paragraph reads a little awkwardly for me. The present tense is a little odd, though I do understand the intent. I feel like the current content could be shortened into a couple sentences and the rest of the paragraph fleshed out a tad.

I agree, I think the following argument reads well without this paragraph and wouldn't lose any meaning if it weren't there.

I'm actually just confused about the purpose of this paragraph. Is it just explaining how to think about divide-and-conquer?

I agree - I think the whole paragraph could be scrapped and there would be little impact lost.

I think this part is fine. It is not that informative, but a textbook that is information dense is too boring. This adds some spice to the whole thing.

I agree, the paragraph doesn't really explain much. It gives a bit of a transition into a better way to represent divide and conquer I believe but it seems a bit unnecessary.

These two beginning paragraphs are unnecessary and awkward. This whole part could be replaced with something like: "Estimation problems can be difficult and complicated to solve, and often have a hierarchical structure. For such problems, a tree structure can be a useful representation."

ˆagreed.

I disagree.. I find in these readings the conversational tone keeps it interesting. I personally enjoyed the personality in the intro.

This paragraph transition is also a bit weird sounding - it definitely makes sense, but I think it could flow a bit better to make the point that the best method of truly representing the estimate is a tree.

track whose 'rings' lie 1.6 µm apart. Along the track, the pits lie 0.9 µm apart. So, the spacing is between 0.9 and 1.6 µm; if you want just one value, let it be the midpoint, 1.3 µm. We made a tasty pudding!

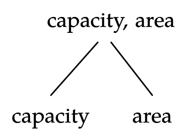
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

What does this mean? It reads very abruptly.

I think what it's saying is that in previous methods, we thought of the estimation methodology as very sequential and simply doing one step after another. What this Tree representation methodology is trying to improve is organizing the different components of our estimate so that it is easier to understand, rather than doing step by step where you could easily get lost.

This was also very confusing for me...I think use of more proper nouns would make the introductory sentence of this paragraph less ambiguous.

Linear and sequential also seem redundant (is there a difference in this context?)

This paragraph helps to introduce the trees and why they might be so useful. I think its a good way to introduce it.

What is the motivation for finding a fitting structure for this estimation process? I might suggest spending a moment saying why we want this whole thing to be visualized in a structure at all before diving into the tree details.

I totally agree with this. A tree diagram would definitely be a more efficient way of displaying the problem. When reading the previous section, I could not remember the process the author went through to get the answer. I will be using this for study sheets now

track whose ‘rings’ lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

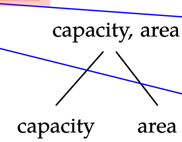
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate’s structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let’s construct the tree representing the elaborate divide-and-conquer estimate for a CDROM’s pit spacing (Section 1.1). The tree’s root is ‘capacity, area’, a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree’s root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the ‘area’ node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

How does this fact bear on the actual procedure used in estimating? Should we write the tree as we work, or produce it afterward?

I agree - when should the tree be created? Is there anything in particular we should be looking for when producing the tree?

I think it’s more the idea of the process. Going along with the recipe analogy, some cooking plans you have ahead of time. People might know for example that pancakes are going to go well with syrup before they are made. Other times you might stumble upon toppings that go well that you may not have expected - cooked bananas for example. You add these to the recipe as you go. Still other times, you might just be trying out new stuff. Then when everything works out, it’s good to document it so you can do it again. I think it’s flexible like that. Now you just gotta learn how to flip pancakes.

I feel like it depends on how you think. I would probably go about drawing the Tree as I worked through the problem. I also feel like how you draw the tree depends on your thinking style. As much as I don’t want to, I can compare it to breadth-first and depth-first search. If I were to draw a Tree, I would do it depth first, working completely through an example and then moving on to the next.

I use a tree as I estimate, but most often I do it in my head. Maybe I do it in my head because I have lots of practice. So, my suggestion is to elaborate the tree on paper as you work. With practice you’ll end up doing lots of the elaboration in your head.

Stating that the tree is the "most compact representation" says that there are other ways of representation. Do those other ways refer to linear and sequential or other representations for hierarchical information?

Are there similar constructs for the other methods we will develop?

For several of them – lumping, springs, easy [extreme] cases – a very useful visual representation will be graphs. I have lots of examples and will share those. I’ll also keep looking for visual restructurings that are as significant as the tree representation is for divide and conquer.

shouldn’t the ’root’ be what we’re trying to find?

track whose 'rings' lie 1.6 µm apart. Along the track, the pits lie 0.9 µm apart. So, the spacing is between 0.9 and 1.6 µm; if you want just one value, let it be the midpoint, 1.3 µm. We made a tasty pudding!

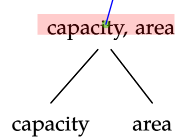
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

Why is capacity, area the root instead of pit spacing? What part of a generic problem will typically constitute the root?

I agree that capacity, area is not a particularly good name for the top of the tree. It adds no information since the two branches are themselves "capacity" and "area". "Pit spacing" gives some explanation of why you might want to find capacity and area, so you don't lose track of what you're trying to do.

Would we normally construct a tree after performing our estimations, or should we construct a tree as we work through a problem?

From what it seems, a tree method is just a way to organize our calculations so theyre easier to visualize. But we'd still have to come up with our whole plan beforehand, otherwise we'd end up drawing multiple trees and then putting them together (which should be ok too, right?).

For me, the tree seems like a helpful place to start. I am happy that this paragraph goes back and addresses a previous example so we see where we could have broken it down.

My guess is it depends on style. If you can see easily what needs to be done, then there is no need to draw a tree. But if you're lost, the tree is probably a very helpful tool to visualize where you could go and what you might be missing.

I think the tree just provides some visual structure with which to organize our divide-and-conquer thought process.

For me, this is something useful for me to do as I go along, even if I do have to revise it. I really learn from diagrams and need the clearer organization... for me this section is more helpful/clearer than the previous ones.

This goes back to the very first paragraph of section 1.1 when he divided the problem into the a) The problem of estimating the area of a CD and b) the problem of estimating the capacity.

I feel like the tree is just a way to visualize the problem in a different way. It all depends on what you're comfortable with and what works best for you

can this same idea be utilized for all divide and conquer problems? and how can you determine if this is a effective strategy?

track whose 'rings' lie $1.6\,\mu m$ apart. Along the track, the pits lie $0.9\,\mu m$ apart. So, the spacing is between 0.9 and $1.6\,\mu m$; if you want just one value, let it be the midpoint, $1.3\,\mu m$. We made a tasty pudding!

Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

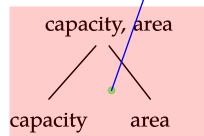
Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.

As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.



Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

Is the main advantage of the tree representation that the pieces of the divide-and-conquer estimations are easier to visualize?

I really like the use of the tree method. I think beginning the divide and conquer explanation with this structure is a great idea and should be put sooner in the paper.

i think so and it's easier to break a complicated problem into multiple parts, which are often easier to analyze

This seems like a more reasonable way of explaining the pro's of redundancy, though seems like it might get tedious if there are many things to estimate.

Shouldn't the root be pit spacing? The two instances of capacity and area seem redundant.

I think the text might be trying to distinguish between other approaches such as "optical" and "hardware" as discussed in sec 1.2, so that if we added a "pit spacing" root node, "capacity,area" could be one if its child nodes.

I agree, but it still seems that "capacity, area" is overly redundant and unoriginal. Perhaps something like "Storage" or the like, which implies both capacity and area, would be better.

It might help if there were some way to flag here at this point that this tree will be attached to one with "pit spacing" as its root (as happens later in the section). For the label "capacity, area" makes more sense in the context of the larger tree than it does on this standalone tree.

agreed

maybe the tree diagram is only used to describe your problem, because now we can take this information and solve other things related to CDs instead of just being limited to finding the pit spacing

track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

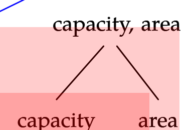
Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.

As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.



Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

So this tree is hanging from the ceiling?

this is standard programming language, and a standard diagram

Typical trees used for analysis typically start from the top down with the parent node at the very top. Another way to do a tress is from left to right. Either way, the point is to set up a hierarchical structure so you can relate certain "leafs" together.

It's referring to a graphical and mathematical, as opposed to biological, tree.

And even in biology, pedigree charts are arranged in very much the same fashion, with the parents forming the root node at the top and children forming nodes below.

This reading clears up a lot of the confusion I had when I was reading the later notes. I guess I was in a hurry catching up to the class that I read some out of order, and that confused me a lot. Now it makes a lot of sense.

Is it the specific example or are we going to use the same names for the root and branches.

i think it is the specific example in this case, however it does appear in a lot of problems i think it's a somewhat general approach

Isn't sprouting capacity, area into capacity and area a bit redundant? Shouldn't capacity and area come from some higher category describing the end goal?

That's a good point - by using two tags that are the categories, that step seems pointless. However, I think in real world use, we could use the 'root' for tags beyond just what the resulting branches are. 'tens' and 'ones' don't have to trace back to 'tens, ones'. I suppose you could call the root 'cd guess stuffs' but that's obviously not very useful compared to 'capacity, area'.

As an additional note, when you get to the end of the lesson, you can see that this practical approach is useful because you may wish to expand your trees later with supersets.

I think the "capacity, area" tag is supposed to help you identify the method used, and the tag doesn't necessarily have to be the combination of its two branches. But it should be more descriptive than "cd guess stuffs" in order to distinguish itself from other estimation methods.

track whose 'rings' lie 1.6 µm apart. Along the track, the pits lie 0.9 µm apart. So, the spacing is between 0.9 and 1.6 µm; if you want just one value, let it be the midpoint, 1.3 µm. We made a tasty pudding!

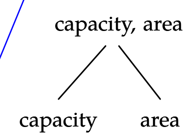
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

How can we decide which operations are trivial and which are not? Because here we are not actually estimating the area, we are estimating the length. I think we should be explicit here that we estimated length (and made it into a square) since that could be a considerable source of error.

it's trivial if you can arrive at an answer mentally. it was originally described as a grid, so that seems to fit the 'area' category fine.

It seems that the leaf can remain a leaf if you do not need to think of new estimations to produce the answer, in this case the only thing you need to estimate is the length and simplify square-ness. If you were to make this a branch, it would only be connected to area –> square simplification –> length; which is not very useful and really just 1 estimation. The other branches from the rest of the section are actually separate estimations. You need the playing time, sampling rate, and sample size (All estimations alone) to calculate the audio content.

I thought it was interesting that the first comment brought up error. Would it be possible to use how many nodes the tree has (and thus how many separate estimations must be done) to roughly estimate the error/exponent associated with our answer?

How? Each problem requires a different number of estimations, and doesn't this bring up the original question of whether one should estimate or leave the node as is?

I'm glad you asked! In the chapter on "Probabilistic reasoning", we will discuss how to estimate the error in one's estimate - and by extension how to choose which leaf nodes are contributing most to the error. With that information, one can plan where to put more effort into refining the calculation.

track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

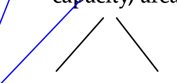
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

why do you think it's easier for us to estimate some values, such as area, than others, such as force?

I think it's the amount of physical experience, plus misconceptions from ordinary language. For example, the ordinary use of 'force' overlaps with the physics concepts of energy, force, and power. So to start with it's hard to have a good estimate of a physics force.

Second, we use area a lot more than we use force. So the fundamental units of area are much more familiar than the units of force. For example, I instantly can see 1 m^2; but 1 N? That I have to think about.

Then I remember a useful rule of thumb: 1 N is the weight of one (small) apple. Perhaps the kind of the apple of legend that fell on Newton's head while he was sitting under the tree contemplating gravitation.

when is it a good plan to utilize intelligent redundant? when is it necessary?

How is this redundancy, if (1) solves it for you right off the bat? It seems beyond redundant, if that makes sense.

I am curious about this, too. Are we assuming that we can't trust the manufacturer? Or are we looking at the box to check our answer? If the box is already available to us, then why are we trying to estimate the capacity using other methods?

It seems like this is an example and you happen to have the manufacturer's information, but if you had another source instead it is good practice to keep this structure even if you are confident in one of the methods used.

I'm fairly certain that the box is implied to be someone's attempt at the answer (possibly a friend's guess). The redundancy is in the estimation.

Now in this tree context I understand this a lot more. and even the idea of intelligent redundancy can be applied in the tree method

I like how it is made clear that a tree can be used to either split some component of a problem into parts, or split it into different ways of calculating the same thing. (intelligent redundancy)

Yeah the tree makes it a lot easier for me to wrap my head around the CD example.

track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

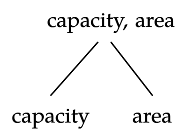
Problem 1.3 Robust addition
The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 1.4 Intelligent redundancy
Think of and describe a few real-life examples of intelligent redundancy.

1.3 Theory 2: Tree representations

Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.



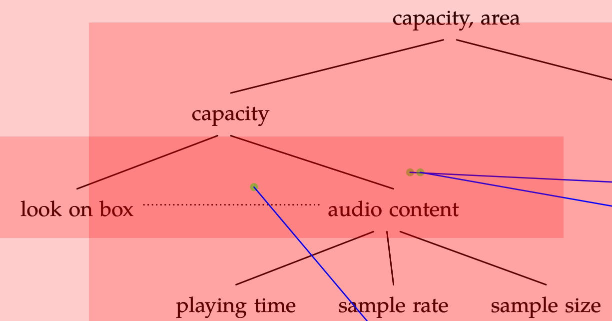
As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (Section 1.1). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.

Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio

I understand the concept of intelligent redundancy, but I have not yet heard a good explanation of what to do when the guesses differ greatly. In all of these examples all of the numbers work out, but what should we do when it doesn't work out so nicely when we are solving problems.

You should probably recheck both and then find an alternative way.

CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:

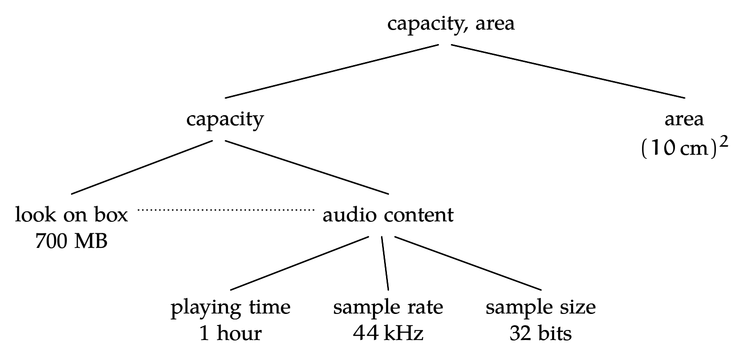


The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see Section 1.2). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;
4. sample size: 32 bits;
5. area: $(10\,\text{cm})^2$.

Here is the quantified tree:



The entire paragraph flows well, moving from leafs to more branches. It seems as though some of the more important vocabulary is very abruptly introduced. I would prefer reading about the idea and then the name.

what is the difference between sample rate and sample size?

I think that the area leaf should have two branches: Estimate and Measure. After all, if you can look at the box to get the capacity, then you should be able to measure the size

these trees would definitely be helpful in explaining what you're doing, to someone else.

this is much easier to follow than the previous sections. i like diagrams that show where we're going.

I agree with the above - the estimations made in the previous statement make much more sense now, in terms of how they go together.

I also really like this tree, and even though you don't explain trees until this section I think it might be nice to see some representation of this earlier on.

Agreed - it might be nice to place these trees in the previous sections to make the reasoning/logic more clear.

could this section even go first? Introduce this as the first section in order to say, "this is how we will solve this problem", and then use the earlier sections to fill in the numbers?

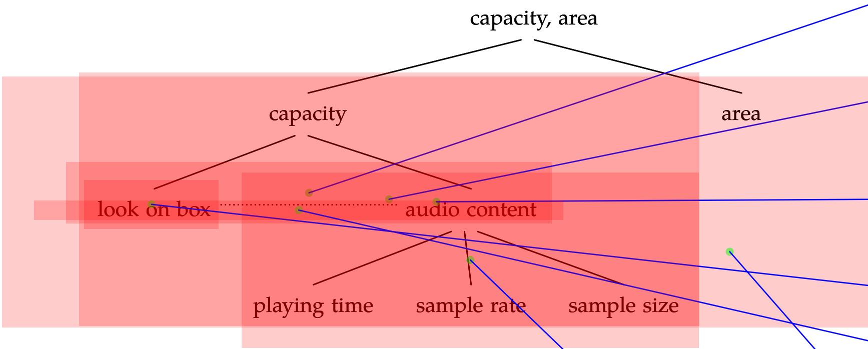
I agree this is a much easier way to understand the concept of divide and conquer. I'm looking forward to the next section to see another example to see how it introduces an idea as opposed to reviewing it.

I agree with everyone, this tree is an excellent example of divide and conquer and illustrates the concept well. I think I would have had much fewer problems with the earlier sections had this diagram and tree explanation been put earlier in the text.

When does dividing become unhelpful and cumbersome? How many methods should we stick to and how detailed should we "divide" them?

so we are no longer at two different portions of the problem now, but rather two different methods solving the same issue?

CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:

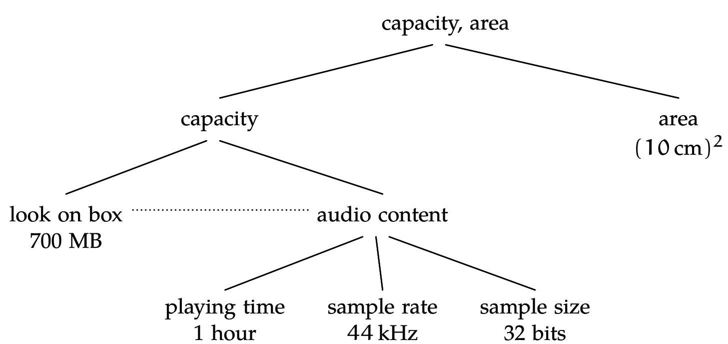


The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see Section 1.2). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;
4. sample size: 32 bits;
5. area: $(10\,\text{cm})^2$.

Here is the quantified tree:



the dotted line is a little confusing. is there another way to represent intelligent redundancy?

Looking at this diagram now is helping me better understand this half of the approximation from the previous chapters. I feel as though this isn't really a separate method, but more of a visual for the previous methods. It's not like this tree diagram really replaces what we had to do before.

This seems like a confusing system – in the top level, the two branches are both necessary to estimate the pit spacing, yet in the second level, either looking at the box or calculating the audio content is necessary. Perhaps we need to distinguish between ANDs and ORs?

"Look on box" doesn't seem to fit the way the other branches are named. This is an action whereas the others are not. Perhaps use "labeled capacity" or "manufacturer's capacity."

Is the dashed line used in refer to redundancy?

Question answered in the next sentence. Oops. ;)

This comment box made the dashed line hard to see, and I didn't notice it until I read the comment. Does anyone know if there is a way to move the comment box outline?

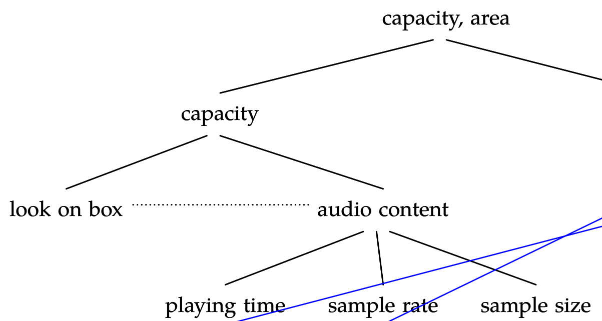
I found you can toggle the visibility of the comment boxes by clicking the little icon of a folder with a paperclip at the top of the Nav tab on the left of the screen.

This graphical method provides a very good way to think about approaching approximation. But are there other aspects we can incorporate into the tree like the soft sides of the calculations?

I'm not convinced that this tree diagraming is consistent. Shouldn't "Playing time, sample rate, sample size" be one node off of audio content (as a method for estimating the audio content), and then playing time, sample rate, and sample size be leaves off of that node? It should be analogous to how "capacity, area" is a leaf off of pit spacing, later, and capacity and area are branches off of that. Because what if we had another method to estimate audio content?

I agree with this, I am a little confused with the notation. I think the bar and leaves should almost be reversed - The bar connecting things that interact to form the parent, perhaps with the operations in which they act detailed on the bar. Separate "leaves" can then describe redundancy. This way things that interact are always connected.

CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:

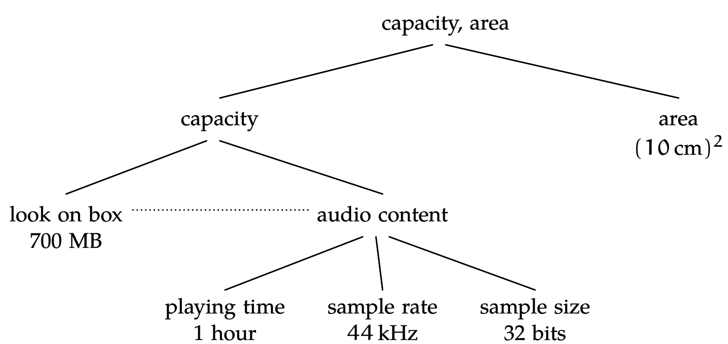


The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see Section 1.2). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;
4. sample size: 32 bits;
5. area: $(10\,\text{cm})^2$.

Here is the quantified tree:



There should be some way to explain how these are put together to find the 'audio content'

does this mean that because we can look on the cd and read how much audio time is on the box, we can figure out how big it is, or because the size of the cd dictates how much audio time is on the cd?

The line is a bit hard to see now with everyone placing comments on the tree.

So a horizontal bar means two different methods of finding the same solution, while if there is no bar both children are necessary to come to the conclusion (the parent)?

Right. I have used tree notation to break down problems before, but I think this idea of representing redundancy on there too is flawed. A dotted line doesn't indicate OR to me.

A related question: Say we had a fourth node under "audio content" and it had a dotted line to "sample size", do we assume that the fourth node is a redundant method for calculating "audio content" on its own, or simply an alternative to "sample size"?

I _think_ it would be an alternative to sample size, whereas multiple dotted lines, or perhaps a different notation, could describe an alternative method of deriving audio content.

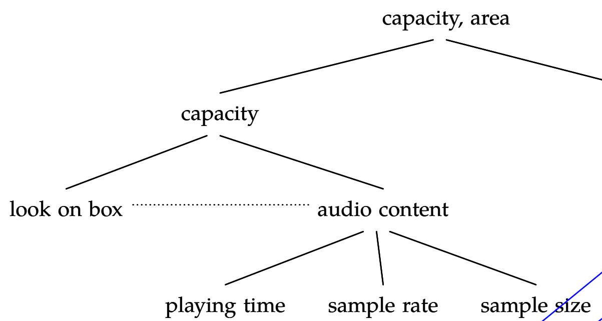
If you wanted to represent alternatives to calculating sample size, would you have two branches coming off of the sample size node indicating the different methods? I think I am confused because it seems like the branches represent two different things – how to divide the main problem into subproblems, and also how to approach the problem in different ways. (I think this is the same confusion from the last section about how redudancy relates to divide and conquer)

I agree. It seems like the tree structure is trying to accomplish too much at once. However, it is also dependent on how deeply we are to use these examples (whether they are to be robust to every situation or are to just designed to convey a specific concept)

Why does it not represent "or"? Rather, if the point is that it is supposed to be redundant - it shouldn't be that you have to choose an "or" but keep in mind both answers to see the robustness of your conclusion. Also, I think it makes more sense to draw 2 branches coming off of sample size, as that is what we did for when we had 2 estimations for capacity.

Also, using dotted lines can get messy when making these tree diagrams by hand

CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:

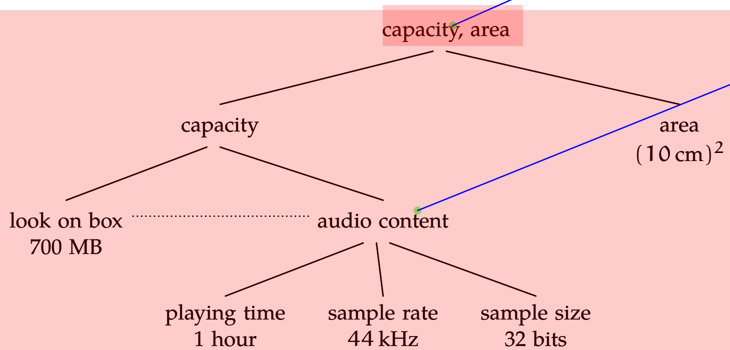


The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see Section 1.2). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;
4. sample size: 32 bits;
5. area: $(10\,\mathrm{cm})^2$.

Here is the quantified tree:



I think the dotted horizontal line is actually making the diagram more confusing.

But this isn't meant to imply that they depend on one another, correct?

Should we also include +/- tolerance values here in the tree so that glancing at the tree gives us an idea of our error?

I think so. If nothing else, this will give us a way to assess the value of the different estimates that we came up with through the different redundant methods

I like this list. It might be helpful to start with a list of things like this even if you don't have a method planned out yet.

I also think this list is nice; at least in the text. it gives a good layout of the 'tree'.

I think everyone wishes they had this list at the beginning, but then it wouldn't be an estimation. I think I'm beginning to understand that a big part of estimation is just determining which method requires which information for an estimation.

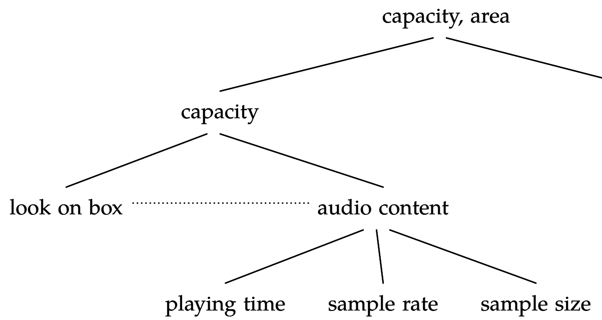
I agree with this above statement, there are so many relevant statistics but determining what is useful is a very crucial part and the tree allows us to start doing this without putting an excessive amount of thought into it

Since audio content depended on some function with playing time, sample freq, and sample time as variables, shouldn't the top of the tree be pit spacing, and then that would branch into capacity and area, since pit spacing is a function of capacity and area?

is there any other way to word "capacity, area". this is around the third or fourth time i've seen this and it still looks as it did awkward as the first time i saw it.

I always have trouble looking through my notes later. I'm going to start using trees when I do the hw problems

CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:

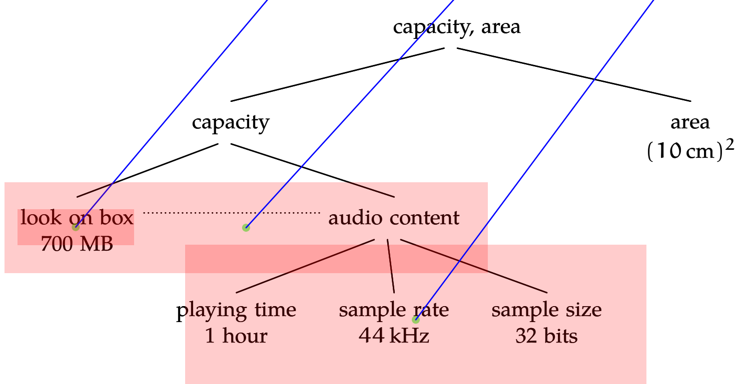


The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see Section 1.2). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;
4. sample size: 32 bits;
5. area: $(10\,\text{cm})^2$.

Here is the quantified tree:



I realize the redundancy is to strengthen the estimate, however, given that one of the values (the 700MB) is a "given" (found on the box) is the other one really necessary? I understand its point if both values are estimates (again, redundancy) but if one is a given value, why would this matter?

I completely agree with that

instead of thinking of it redundancy to check, could we instead just treat it as having multiple methods to tackle the problem. for example, if the box is lost.

Is this being over-redundant, or in general should we not trust this kind of data from the supplier?

This tree is really helpful but I still don't understand how I would know that audio content breaks down into these 3 components. In a problem, would this be defined for us or is this something that would be regarded as common knowledge?

Yeah, this I don't think I'd split that into 3 different leaves.

It may not be common knowledge before reading the section, but hopefully afterwards it is!

To answer more specifically: It depends on the level of the question. An easier question, might give you the breakdown and ask you to estimate three numbers. A harder question, more like real life, might leave you to figure out any way that you can. Both questions have their value (one cannot play Bach without first learning to play scales).

The overall concern is a big reason for my grading system where correctness is not relevant. Don't worry about doing 'badly' on a question. Just keep your eyes on the prize: If you are diligent in this course and continue to practice, you'll get fluent doing all of this. So, just aim for that.

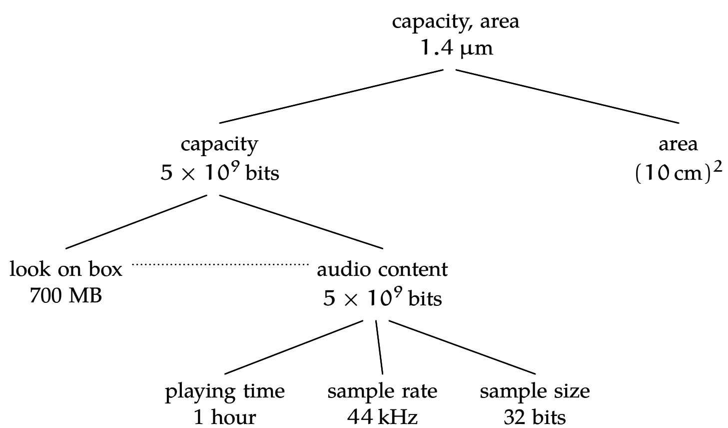
The final step is to propagate estimates upward, from children to parent, until reaching the root.

▶ *Draw the resulting tree.*

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

Why change the vocab to children and parent? Is this a more general way of explaining trees or just an analogy?

The vocabulary of children and parent is widely-used and accepted when referring to a tree data representation. Although I do agree, the change from root/node/leaf to parent/child is odd.

I agree it is confusing, but it's okay here. In fact, it's a good way to introduce vocabulary.

This note is very well organized and easy to follow.

What do you mean by this? You already gave us the tree...or do you mean, 'fill in the resulting 'non-leaf' nodes?

I would like to see this somewhere on the tree, even if it was just an 'x' between each of the 3 leaves to show that they are multiplied to get back to the earlier branch.

it gets too cluttered that way, and you lose the benefits of having a tree in the first place.

maybe instead of drawing the symbols between the leaves, making the tree more cluttered, it would be useful to draw the operation symbols between the branches

I agree–some way of seeing how the branches combine to form the parent node at each level has been helpful to me when I draw trees. I think the tree is most helpful for conceiving of the dependency between parameters, and nearby or separate equations can explain the mathematical relations between them.

The relationship might not always be multiplication. And we might not know what the relationship is. But using the tree, now that we know what the dependent parameters are, its easier to formulate a relationship.

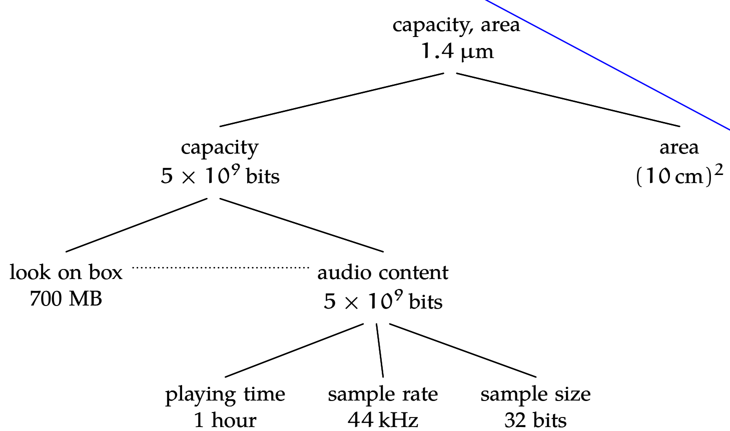
The final step is to propagate estimates upward, from children to parent, until reaching the root.

Draw the resulting tree.

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

Draw a tree that includes these methods.

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

how do you know it's the product of these three variables? Is there a formula for this?

I think it's product as in derived from separately, not actually all three numbers multiplied together.

I think it's that the way they set up this tree, it happens that playing time*sample rate*sample size gives audio content in bits if you do unit conversions for Hz and hours–in the first reading this was how they did it too, it's just much more clearer now using the tree

Exactly, multiplying hr*KHz*bits=bits, thus we used unit analysis (dimensional analysis).

ok- but this is not the general case, right? like multiplying sibling leaves isn't always the operation necessary. Is there a way to represent the operation performed on a set of leaves in the tree?

I don't think that is the general case. I think depending on the problem, the necessary relations will vary. The tree does a good job of breaking down the problem into manageable pieces (divide and conquer) however, it does not provide much insight into how to derive a solution from all of these broken up pieces. Unless someone else has seen this. thoughts?

perhaps there could be a legend on the side, it could clearly state that the dotted horizontal lines show redundancy

It seems strange that they found exactly the same answer. The numbers 44hz and 32 bit sampling seem too accurate to have been guesses, unless the guesser happens to know that they are the real values, I don't think the average student would be able to find this answer with such accuracy.

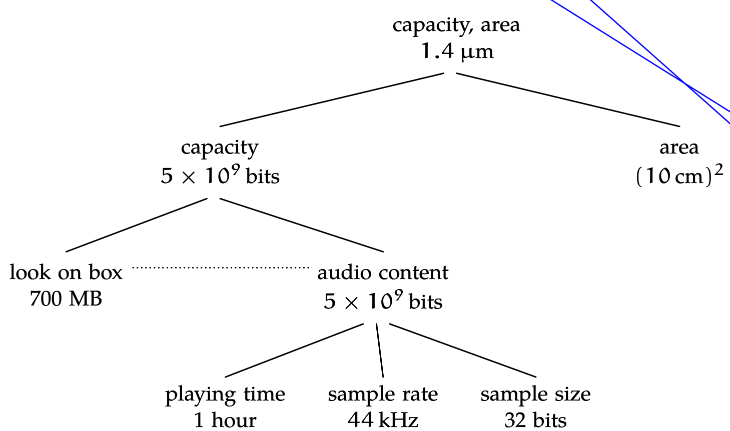
The final step is to propagate estimates upward, from children to parent, until reaching the root.

▶ *Draw the resulting tree.*

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

If they didn't agree, how would you represent this on the tree?

by erasing the wrong one? (kidding) If they didn't agree closely enough (enough being a key word), then you would probably have to reevaluate to see which one is in error, or find a third method to serve as a tiebreaker?

Hopefully they wouldn't vary by too much. However, we may often have an implicit reliability with each method. Surely the box is more reliable than our estimation via pits. In which case, we would have to use our gut to determine if we choose one over the other or pick a value in the middle closer to one than the other.

Would this be where we could add in the tolerances to our estimate? If the look on box and audio content vary from each other, could we pick a value between the two and set a tolerance based on how far apart they are?

Some people may not understand how 700MB = 5*10^9 bits. Maybe a quick demonstration or mention of the conversion factor may help to make this material clear to the non-course 6 students?

I agree with this I had to refresh myself with this conversion as well, and it might be nice just to see it there instead of looking it up.

I just learned it so I will post it here. Bascially, to convert megabytes to megabits, you multiple by 8. Then to convert megabits to bits, you multiply it by 1048576. Thus 8*1048576 results in 5.8E9, or approx 5E9.

1,048,576 = 2^20

I think a bit of background of bits and maybe even a section on estimating binary representations of numbers could be quite interesting for this class

these formulas should probably be included on the tree somewhere. It would make the entire tree much easier to read and follow.

It would be nice if to the left or right of the tree, the different equations used to go up a node were shown

I agree, the tree shows what are the necessary calculations/number needed but does not indicate how they are necessarily used. In the example here it's not a big deal but I'd imagine in more complex examples the formula written into the tree would be handy

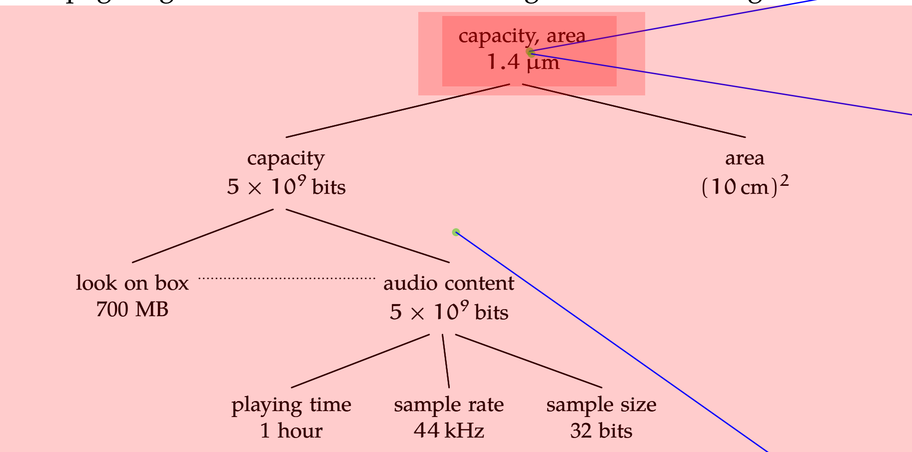
The final step is to propagate estimates upward, from children to parent, until reaching the root.

▶ *Draw the resulting tree.*

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

I believe this is a large problem that may need to be looked at (or hopefully we will look at later), and it is that this does not seem like common knowledge. Although we can find all of this information, where can we get these equations from? Can we better estimate it if we don't know? I understand the concept that we may be doing estimations more in our field, but what if someone asks me this, and I'm not CS, and they want a legitimate answer? I feel like this divide an conquer method is effective in broad cases, like maybe the example on the next memo, but in technical cases, I can't see a strong benefit, though I like the idea of it.

This is a very easy to follow and clear demo of divide and conquer. Is there a way to incorporate/tell how to mathematically process the tree (i.e. how to know whether to multiply, divide, add etc)?

It also might be helpful to have the variable we are solving for (pit spacing) at the top of the tree above capacity, area. I'd like to see it there just as a reminder of what the end product is.

Oh didn't see this comment - I posted a new comment with the exact same opinion. I think this would make the formatting of the tree more consistent.

Given that the number we place here is the spacing in between pits, shouldn't the tree's top level be "pit spacing" instead "capacity, area"?

I believe the point is that the number given is the pit spacing as calculated using the capacity, area method.

However, I agree with you and all the others that perhaps it would be best to start the initial introductory tree with "pit spacing" because it has caused so much confusion so far, and it is clear that most people understand the final tree and are just getting hung up on the fact that we are introduced to a tree that doesn't start at the true root of the grand final tree.

The whole tree is shown on the next page, I think that clears it up.

oh, yeah, the next tree clears it up a bit more, perhaps having a legend would clear up more confusion since some same height leaves have a different relationship than other same height leaves.

This is definitely helpful if you don't get all the way through the problem and need to come back to it latter.

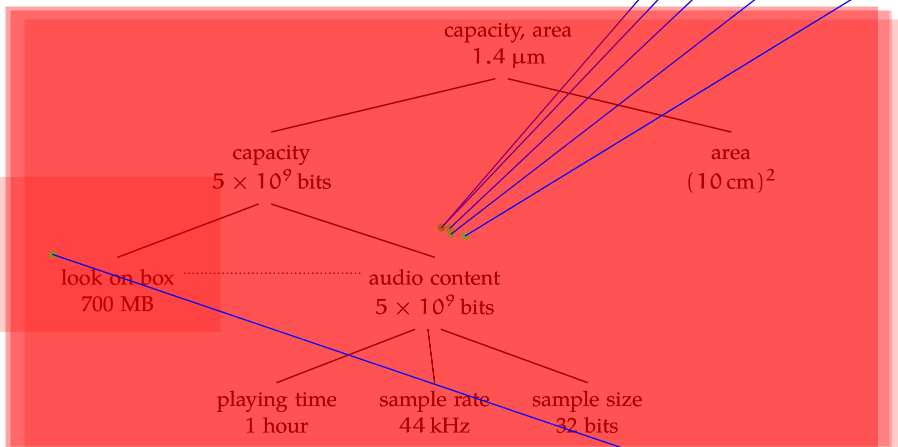
The final step is to propagate estimates upward, from children to parent, until reaching the root.

Draw the resulting tree.

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

Draw a tree that includes these methods.

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

These charts are very helpful, maybe they should come in the earlier sections?

These charts are very helpful, maybe they should come in the earlier sections?

I agree that it's difficult without the various operators so it would be nice to see on the lines what the mathematical relationship is.

I thought the example of MIT's budget today in class left a little more to the imagination on the tree's and was more open....u can get stuck into estimating many branches and going deep down in the tree...when in reality trees should be a fast way to do the analysis

Looking at this tree the original person can understand it, but without the formulas no one else could

Definitely true. Although this makes the concept of how the estimations are broken down clearer, it gives no indication of how to put the same estimations back together and find the answer.

Maybe the "branches" should include a process as to how the answer was reached from the given information. This would make the lines more literal pathways between raw information and the conclusions drawn from it.

I agree that it would be useful, although I think the point of the tree is just to concisely breakdown the steps and numbers and leave out detailed explanations

I also agree. Simple equations could be added between branches to indicate how the solution above was generated. Looking at the chart, it isn't obvious that playing time*sample rate*sample size=audio content.

I was confused at first, but after you study the tree a little it does become quite apparent. Realize that these numbers aren't going to be trival to many people and they are going to have to think about what is going on.

The tree is made to show all the pieces needed to solve the complex problem. It is only a way to sort the info. I think it was a stretch for the author to say that you propogate estimates from parent to child so matter of factly.

So where does it show whether you chose to use the audio content data, or the look on box data?

more compact, yes, though inferior in explanation, right?

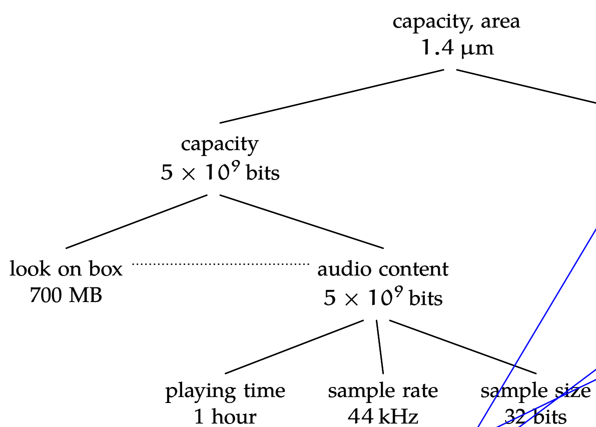
The final step is to propagate estimates upward, from children to parent, until reaching the root.

▶ *Draw the resulting tree.*

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

As a final way to put together all of the information, it would be nice to see the all of the calculations multiplied together to form the result.

Definitely. The tree itself is a great summary, given that we're reading this over the span of a few days, and perhaps the actual calculations (with units) would help reinforce what we've been doing.

i'm a little confused here. is the tree supposed to be a "new" way to solve these problems? because it seems like just a different visual representation of the same approach.

I think the tree is simply a new way to display the information and how the approximation is split up into different components.

I couldn't agree more, which is why I think this should be introduced with the material from the beginning to make it easier to understand.

I agree with this sentence and I do believe trees are a lot clearer. They also give the person who examines trees time to figure out how you got the answer by themselves. Which is definitely an improvement on learning rather than just believing a answer given by a professor must be correct.

How much variation is acceptable between multiple trees for using the same method of estimation?

It seems like we could help account for variation (horizontally), and make the robustness clearer, by adding the (+/-b) to each of the leaves. (I am referring to the estimated accuracy of each value's magnitude as displayed in the exponent.) Though, this may just have the effect of cluttering the tree diagram...

By the end of this section I've forgotten what we are trying to approximate in the first place. It would nice to have a reminder after we have done all the little estimations.

agreed.

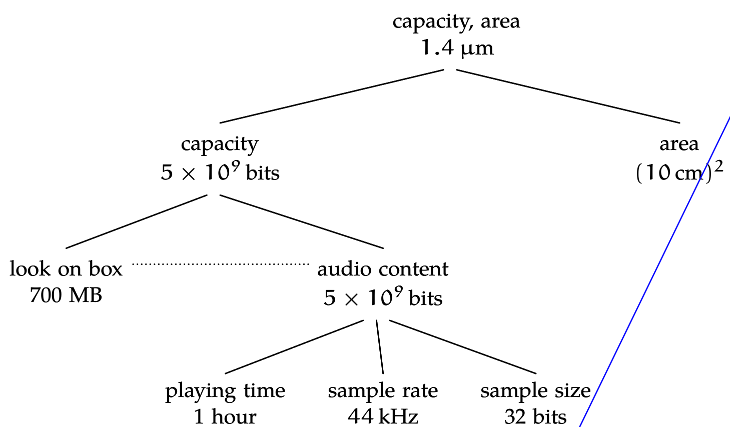
The final step is to propagate estimates upward, from children to parent, until reaching the root.

▶ *Draw the resulting tree.*

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly 1.4 µm.

Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in Section 1.1. The comparison becomes even stronger by including the alternative estimation methods in Section 1.2: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends

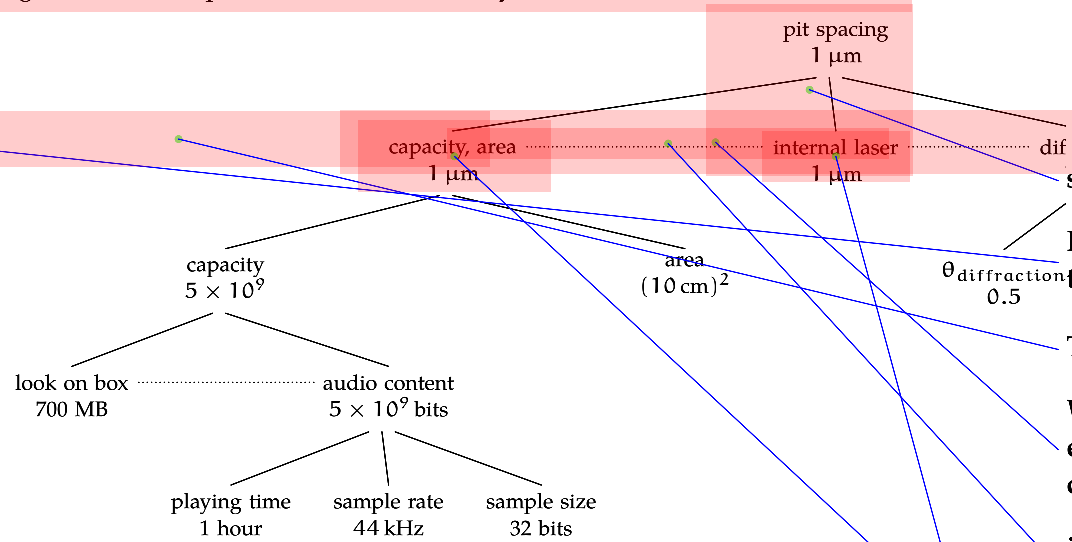
This sentence reads awkwardly- is it supposed to say "depends on just ONE quantity"?

good catch, probably!

I also got tripped up here. I might remove the word quantity altogether and just state "depends on the wavelength of the light"

(The sentence needs "ONE", as noted in this thread.) I remember debating this suggestion mentally when I first wrote the sentence. The reason I didn't phrase it that way is that I want to emphasize the pattern that one quantity means one node. If I mention only the wavelength, it's harder to generalize from this example to the pattern.

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

In my experience, sometimes I may need to come up with individual trees as I get information, then once I have everything, i put all the pieces together like a puzzle. I'm not sure if thats the best way to do it, but thats the best method I know of.

Yeah, I agree–because we don't always have our whole plan written out. I guess in this case it would work, since as we move up the branches, everything is simply multiplied together, so separate trees made earlier could be multiplied together

should this larger tree be written down before we begin to crunch numbers at all?

I think that it's ok to use 'capacity,area' here because there are other methods, however the other tree should still have pit spacing because that is what we're trying to find.

This should be 1.4 because that it what you found it to be on the previous page

Would it make sense to arrange these in order of easiest to hardest to work out, or rough estimate to more known quantities? For example, the internal laser is a rough guess, while diffraction and capacity/area are much cleaner

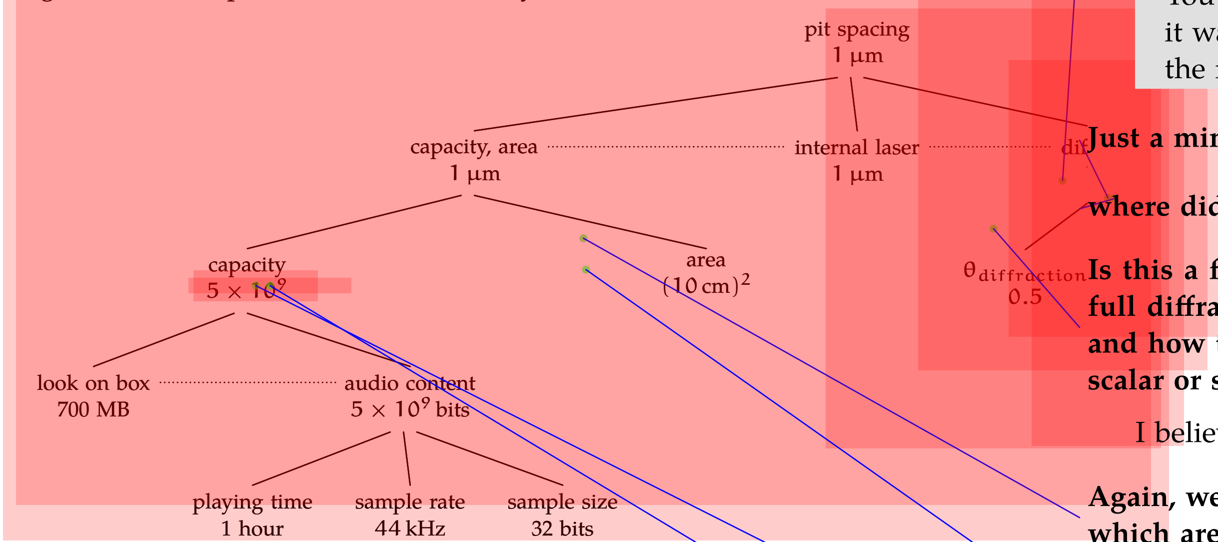
it would seem the redundancy line refers to data that has the same value regardless of dependency. Clarified from earlier

So I made a comment about this section earlier saying that this should say "pit spacing" but now I see why it doesn't - I'm no longer quite sure how to address this but something should be done to make it slightly less confusing. At first glance, relative to the rest of the tree, I would think that this section means "capacity, area" = 1um, which makes no sense.

I was unclear about this also. I thought it should have been pit spacing like you. Seeing as it is not, though, I'm confused as to why the "capacity,area" node needs to be there. Couldn't "capacity" and "area" simply be direct children of "pit spacing"?

I like this one, nice and simple.

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

How much oil does the United States import (in barrels per year)?

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

i'm assuming this just got cut off by the margins?

You're right. (It used to fit on the page when it was typeset on 8.5x11, but even then it was beyond the margin). I haven't yet figured out how to redraw the tree to fit in the margin and still be clear.

Just a minor comment: this area of the tree got cut off when I printed it.

where did this part of the tree go?

Is this a formatting error or is there some other reason that the tree cuts off without the full diffraction example shown? I'd like to be able to see the two children of diffraction and how they actually interact together - if its the multiplication thing, a division with a scalar or sqrt...?

I believe Sanjoy answers that a couple of questions above you. Formatting error.

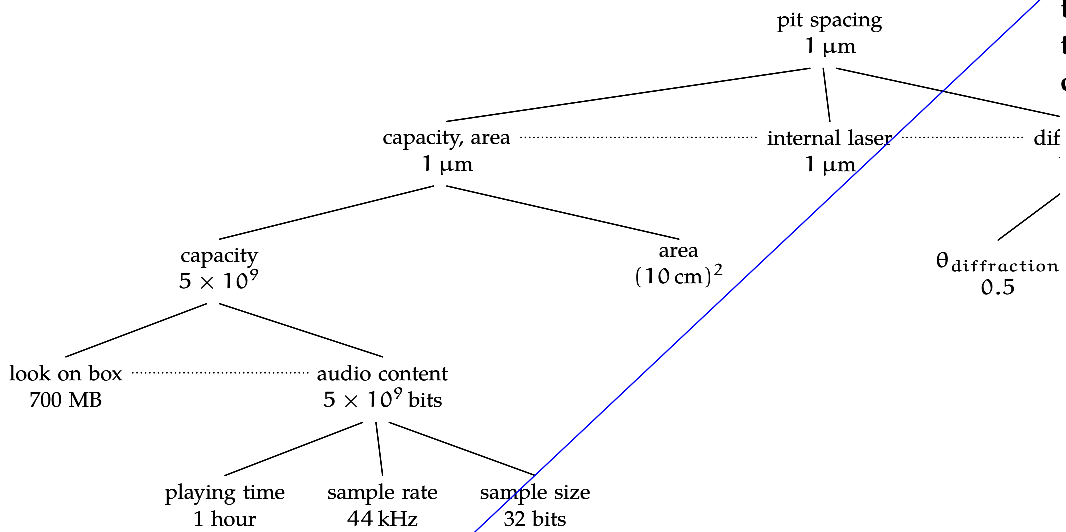
Again, we really seem to need a way to distinguish redundancies from multiple branches which are all required.

Can anyone see this entire tree? I can imagine what should be on the other side but I'm just wondering if this is cut out for everyone.

units = bits

Why don't you keep the units on this version of the tree? Shouldn't you indicate bits?

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

The tree leaves me wanting to know what the equations are that lead up from the leaves back to the roots. There must be a way to include an equation along the sides of the branches to show how on goes from sample slze back to audio content, especially for these calculations that contained a lot of new information for most of the class. While this might make for a nice graphic summary, I feel like it doesn't really have that much content in it.

That's an interesting idea. The most common equation for each node is "multiply all the children to get the parent." But not always. For example, pit spacing is not capacity * area but sqrt(area/capacity).

Even so, the operation is usually multiplication of the children after raising each child to a particular exponent. Acompact representation for that would be to place the exponent along the line connecting the parent and the child. For example, the line to the capacity node (that says 5*10^9 bits) would have -1/2 on it. Similarly, the line ot the area node would have 1/2 on it.

In cases where that framework is not general enough (e.g. reaction rate might be e^-1/T), one could give the explicit formula at the node.

What do you think?

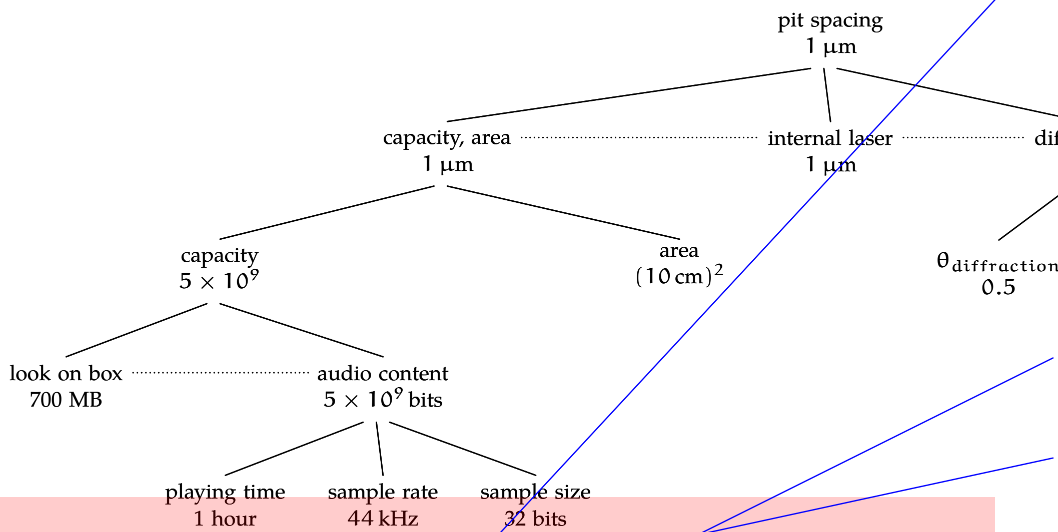
That sounds like it would work pretty well, as long as there was a table of symbols or something that reminded the reader what the 1/2s meant.

That sounds like a simple way to include information about how errors in the estimations for values lower-down in the tree would propagate upwards. It would also lack any constant coefficients present in most equations that are a product of variables, but that's not really relevant to error propagation.

I like the idea of having the equations at the nodes; I'd say that it probably defeats some of the purpose of the estimation, but if you have no equation (no previous knowledge), then whether you can estimate some numbers doesn't really matter. I can agree with that,

Yes, I think that would be a useful, simple, addition to the tree structure.

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

I agree that this visualization makes everything much easier to understand, so perhaps it could have been used earlier on and integrated into Sections 1.1 and 1.2 instead of having a separate part by itself?

I agree that the visualization in the form of a tree is very simple, making the divide and conquer method more clear. However I disagree that it should be shown in the previous two sections because it is a good way to summarize the previous approximations and bring everything together.

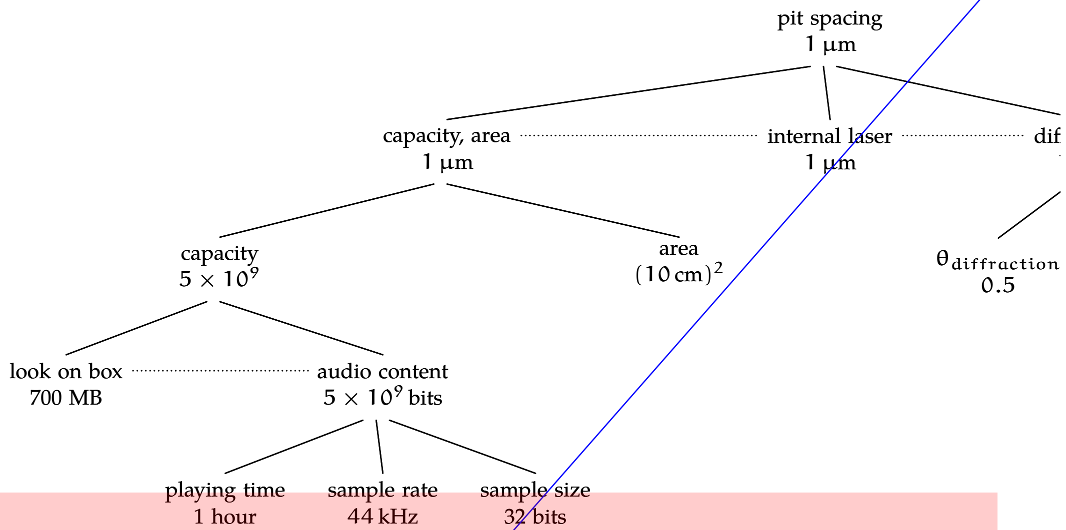
I also think that the tree is placed best as a summary. While it would be helpful to have this tree from the start, we should be allowed to go through the mental process of creating our own version of this tree ourselves so that when it is shown later (like in section 1.2) we can match it with what we had envisioned

typo: makes

Overall, I think some of the less obvious points, such as the operations being done, could be put onto the tree. Having said that, the tree method works very to show exactly how the answer was found in a very systematic way.

Exactly. This is why I think it should come first, it is much less complex and easy to understand. Then you can elaborate into each branch in the next sections.

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

For large trees, is there a strategy of what to expand first? (i.e most complex branch, branch in which you think there are most layers_)

I have a similar question. from what it seems, the tree diagram is basically an easier way to visualize the estimations we are making to arrive at our one large estimation. But the tree doesn't do much more–we can't get from it any new insight as to where to start or how to calculate certain things. So even with trees, we're still pretty much on our own in terms of figuring out what values to estimate and multiply together, etc.

It seems to me that it would be most beneficial to expand the branch with the most numerical calculations, and double check your answer with more other practical methods such as diffraction.

I was wondering about this also. When doing these problems, I usually don't have it so mapped out in my head. The trees are super useful in recalling the process, but I'm not sure how much they would help me solve it.

It seems to me that the tree should guide your estimations. If you reach a leaf which seems difficult to estimate immediately, then look for a way to divide it into a set of easier calculations. Once you've reduced your problem to a set of simpler calculations then work backwards to estimate your desired quantity.

The tree is a great visualization tool, however I like knowing where all the numbers came from. The analysis in Section 1 should be introduced after the tree because it would sink in faster with the picture of the tree in your mind.

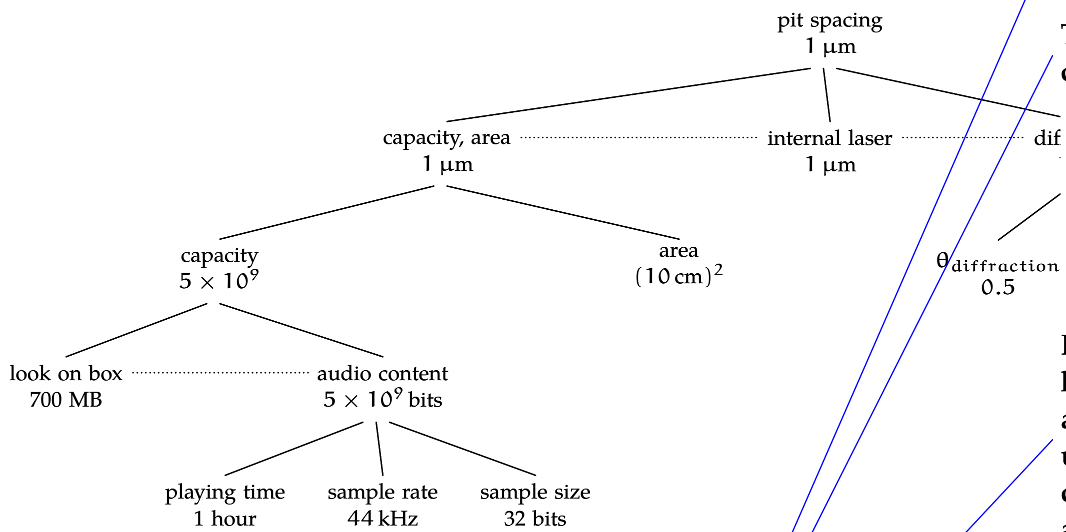
I agree entirely. It makes the series of steps taken to solve the problem much easier to understand and follow.

If we had made trees throughout the sections, thus building up to this intricate and multi-tiered tree, I would have been able to synthesize the previous sections as we completed them. Making these connections makes it very clear why this is a method of divide and conquer and why their is intelligent redundancy.

I thought this entire section was very well written. Simple and concise. I can't think of any ways to improve it.

This was easy to follow, but at the same time, I don't know if I could just come up with all of this on my own.

on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of Section 1.1 and Section 1.2 – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

1.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities:

- estimate how much oil is used every year by cars;
- increase the estimate to account for non-automotive uses; and
- decrease the estimate to account for oil produced in the United States.

Here is the corresponding tree:

The overview definitely helps and I feel shows the consensus among different methods and a similar conclusion - are there any cases where you see a slight difference and end up having to normalize it in a way?

This paragraph summarizes the only use for trees. Once you understand the problem, you can write it as a tree so you can go back and easily understand what you did before.

That negates the notion of drawing the tree as you go along in the problem. I agree with that - I would have been able to initially figure out the approach and organize it into a diagram. Trees are just a way to easily convey information once you know what you're doing.

*wouldn't oops

If you hadn't given me this method to subdivide the problem, I would have no idea on how to solve the problem! With the some help from Google. I found: 20 barrels of oil are used by an average car per yr, there 200million cars in the US, about 66% of US oil is used in transportation, and the US produces 2000million barrels/yr. My calculations come down to being: ((20 * 200M)/(.66))-2000M approximately equal to 4000M = 4billion barrels a year

When we use each method separately, should we automatically add a factor to compensate for the other uses of oil that we are not accounting for in our tree diagram?

I think that each of these quantities have its own branch in the tree diagram.

I thought this memo was straightforward, so there weren't many comments to make.

I thought that I put these comments on the page before, whoops. Over spring break i went through and tried to find any other readings that my comments didn't make it from my paper to the nb system. thanks.