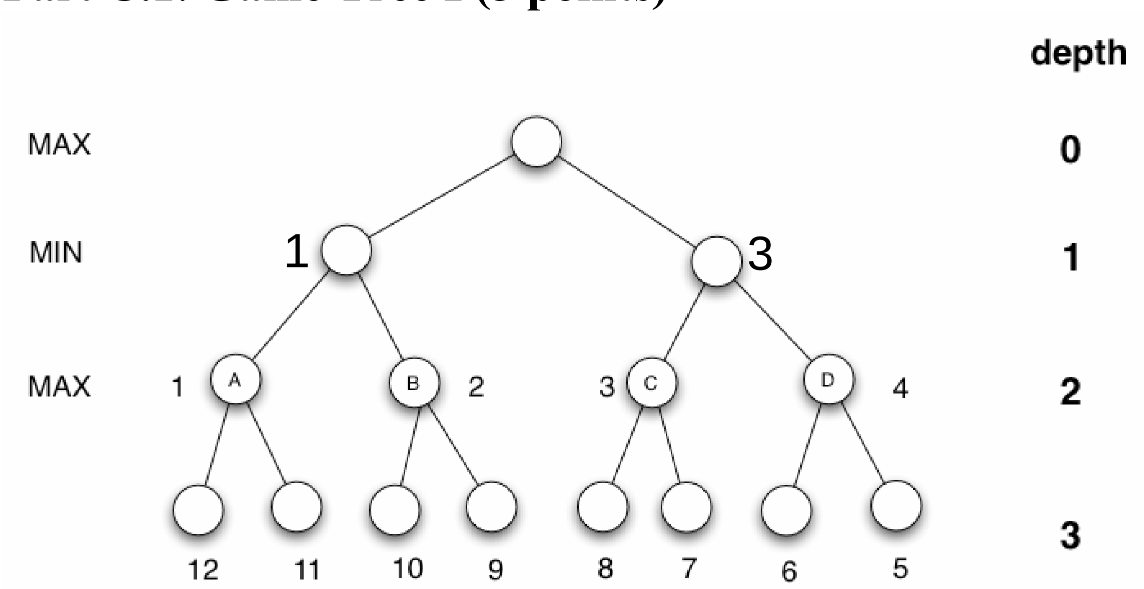
10/13/11 Solution: Minimax with Alpha-Beta Pruning and Progressive Deepening

When answering the question in Parts C.1 and C.2 below, assume you have already applied minimax with alpha-beta pruning and progressive deepening on the corresponding game tree up to **depth 2**. The value shown next to each node of the tree at depth 2 is the respective node's static-evaluation value. Assume the procedure uses the information it has acquired up to a given depth to try to improve the order of evaluations later. In particular, the procedure reorders the nodes based on the evaluations found up to depth 2 in an attempt to improve the effectiveness of alpha-beta pruning when running up to depth 3.

We want to know in which order the nodes/states A, B, C, and D in the game tree are evaluated when the procedure runs up to depth 3, after running up to depth 2 and reordering.

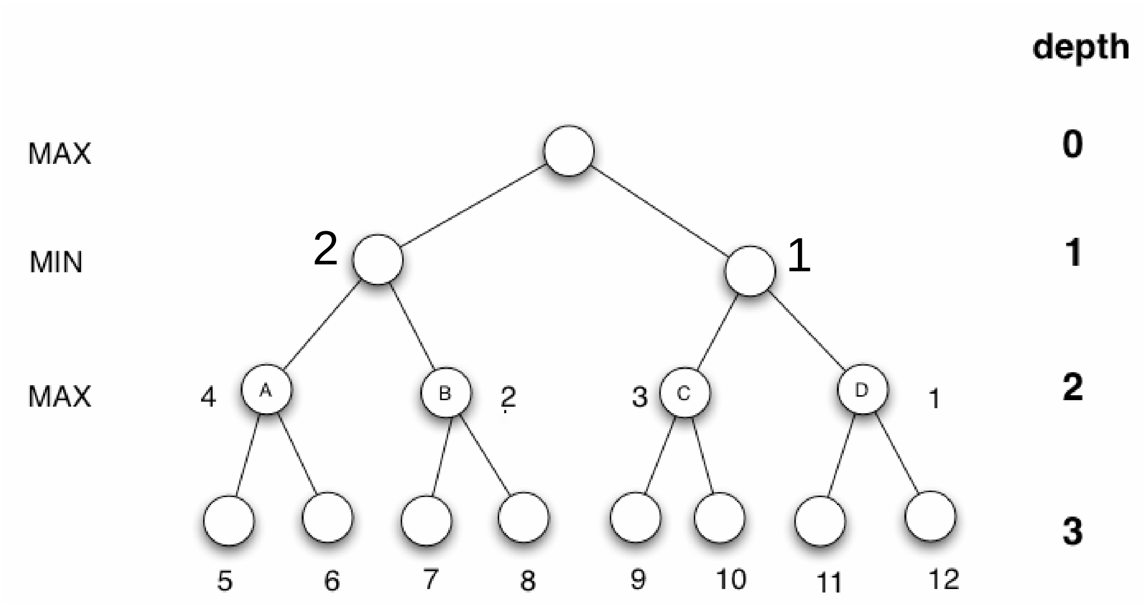
Part C.1: Game Tree I (5 points)



Choose the order in which the nodes/states A, B, C and D in game tree I above are evaluated when running minimax with alpha-beta pruning and progressive deepening after running up to depth 2 and reordering. (Circle your answer)

a. A B C D
b. D A B C
c. B A D C
d. C D A B
e. D C B A

Part C.2: Game Tree II (5 points)

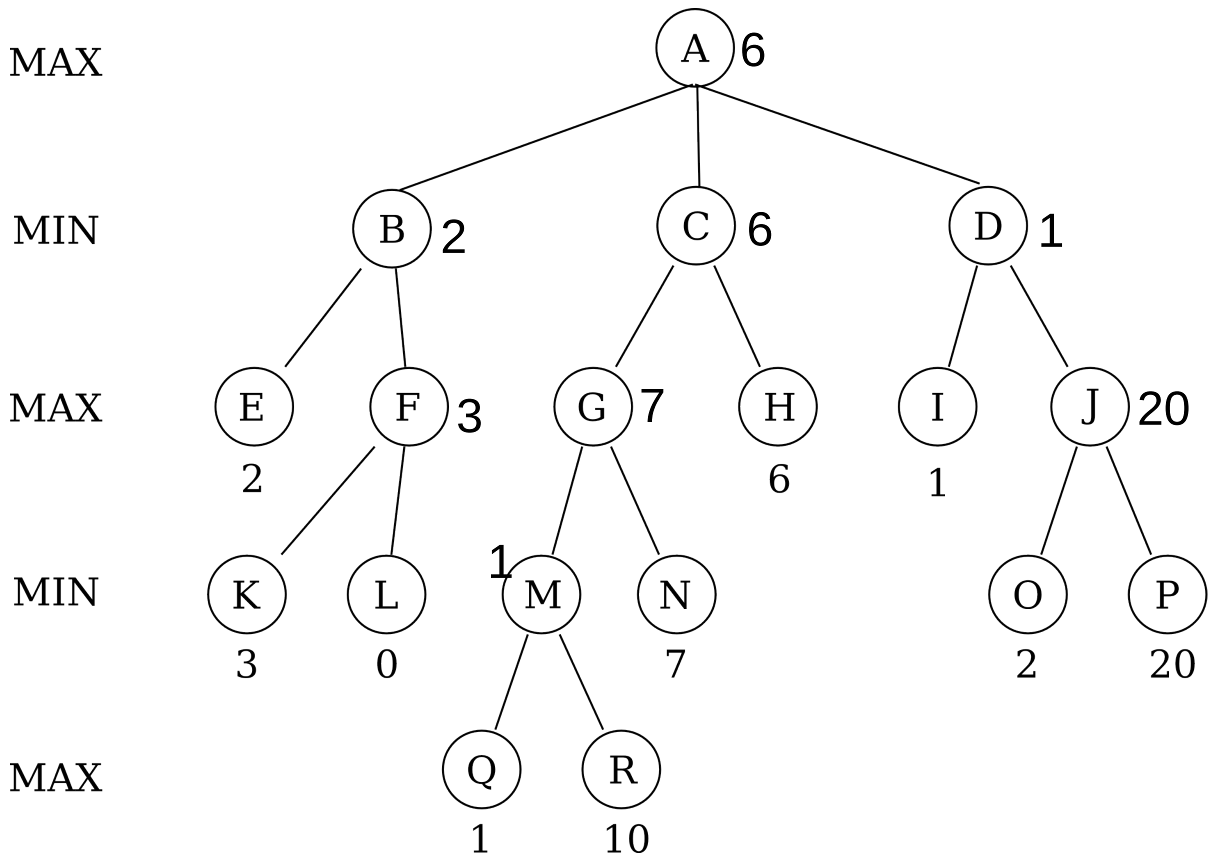


Choose the order in which the nodes/states A, B, C and D in game tree II above are evaluated when running minimax with alpha-beta pruning and progressive deepening after running up to depth 2 and reordering (Circle your answer)

f. A B C D
g. D A B C
h. B A D C
i. C D A B
j. D C B A

10/13/11 Problem 2: Games

In the game tree below, the value below each node is the static evaluation at that node. MAX next to a horizontal line of nodes means that the maximizer is choosing on that turn, and MIN means that the minimizer is choosing on that turn.



Part A (10 points)

Using minimax without Alpha-Beta pruning, which of the three possible moves should the maximizer take at node A?

C

What will be the final minimax value of node A?

6

Part B (10 points)

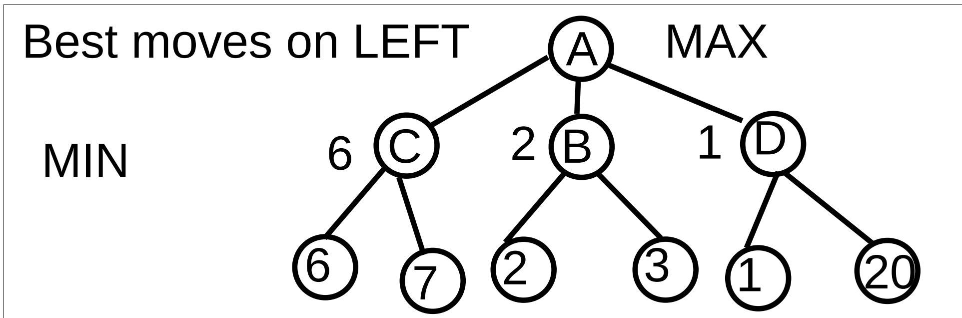
Mark suggests that Alpha-Beta pruning might help speed things up. Perform a minimax search with alpha-beta pruning, traversing the tree, and list the order in which you **statically evaluate** the nodes (that is, you would start with E). Write your answer below. **Note that there is, at the end of this quiz, a tear-off sheet with copies of the tree.**

Not done in this example

Part C (10 points)

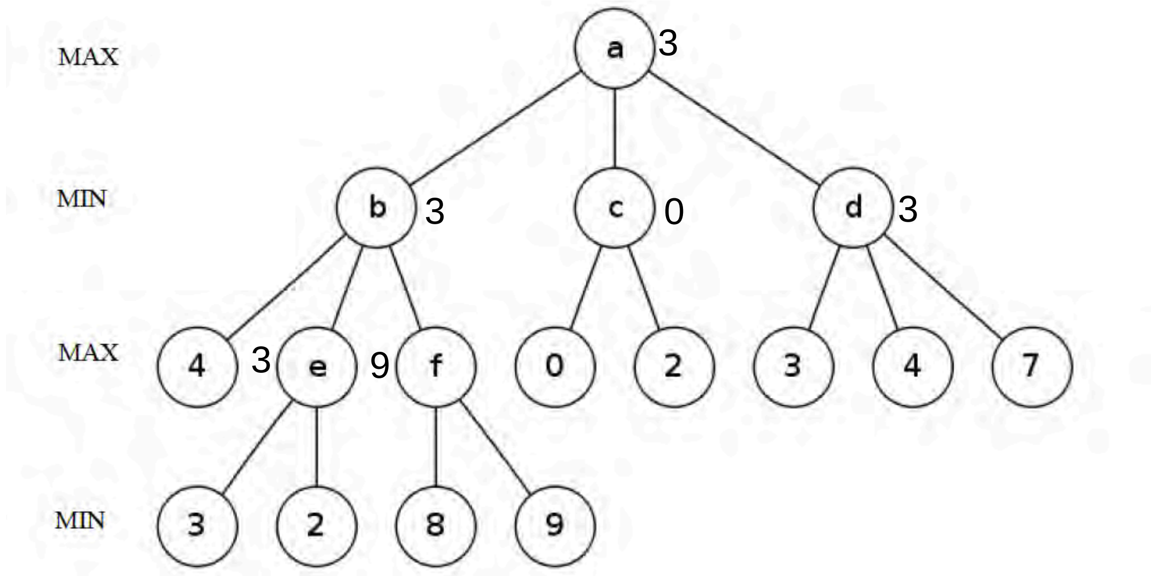
Tom thinks that he might save some trouble by calculating static values at depth 2 and then using those static values to reorder the tree for the alpha-beta search that goes all the way to the leaf nodes. Thus, Tom is attempting to deploy alpha-beta search in a way that will improve pruning.

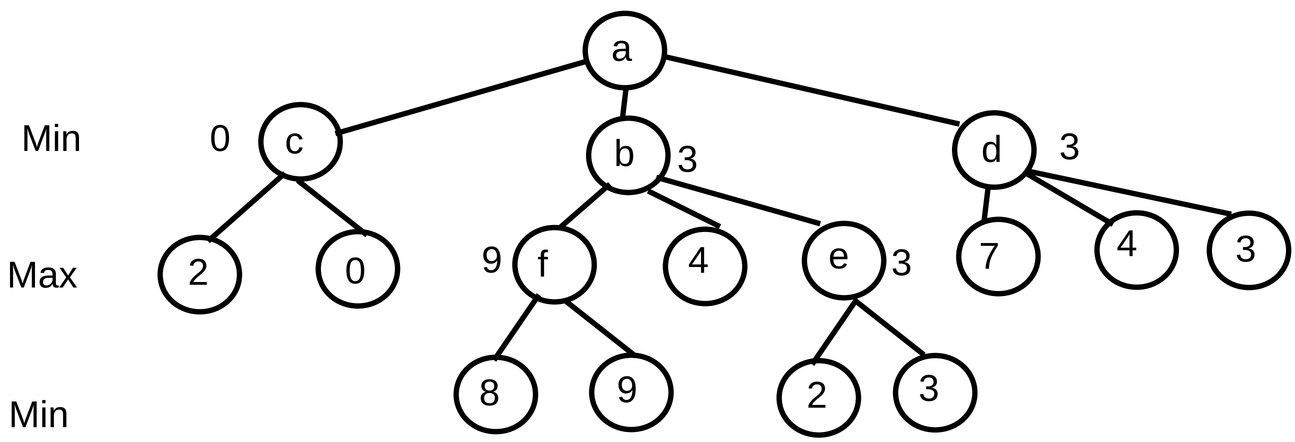
Suppose the static evaluator Tom uses at depth 2 produces exactly the minimax values you found in Part A. Tom uses those numbers to reorder BC&D, EF, GH, and IJ. Note that Tom knows nothing about how the tree branches below depth 2 at this point. Draw the reordered tree down to depth 2, the EFGHIJ level.



Explain the reasoning behind your choices:

Progressive Deepening Games Problem 3





2. Converting problems into constraint propagation form

“Paul, John, and Ringo are musicians. One of them plays bass, another plays guitar, and the third plays drums. As it happens, one of them is afraid of things associated with the number 13, another is afraid of Apple Computers, and the third is afraid of heights. Paul and the guitar player skydive; John and the bass player enjoy Apple Computers; and the drummer lives in an open penthouse apartment 13 on the thirteenth floor. What instrument does Paul play?”

How can we solve this problem? Try it yourself first, by any means you care, then we’ll see how to do it by – ta-da! – the magic of constraint propagation! (You might want to think about constraints when you solve it, and what the constraints are.)

What are the constraints? How might they be represented? We want to use the facts in the story to determine whether certain identity relations hold ore are eXcluded. Here is our notation: assume X(Peter, Guitar Player) means “the person who is John is not the person who plays the guitar.” Further, this relation is symmetrical, so that if we have X(Peter, Guitar Player) then we also have, X(Guitar Player, Peter). In this notation, the facts become the following (of course all the symmetrical facts hold as well):

1. X(Paul, Guitar Player)
2. X(Paul, Fears Heights)
3. X(Guitar Player, Fears Heights)
4. X(John, Fears Apple Computers)
5. X(John, Bass Player)
6. X(Bass Player, Fears Apple Computers)
7. X(Drummer, Fears 13)
8. X(Drummer, Fears Heights)

Now we can represent the possible relations implicitly by means of entries in tables, and use constraint propagation. An *X* entry in a table denotes that the identity relation is excluded, and an *I* denotes that the identity relation actually holds. We can then use three tables, one to represent the possible identities between people and instrument players; one to represent the possible identities between people and fears; and a third to represent the possible relationships between instrument players and fears.

1. X(Paul, Guitar Player)
2. X(Paul, Fears Heights)
3. X(Guitar Player, Fears Heights)
4. X(John, Fears Apple Computers)
5. X(John, Bass Player)
6. X(Bass Player, Fears Apple Computers)
7. X(Drummer, Fears 13)
8. X(Drummer, Fears Heights)

	Instrument Player		
	Bass Player	Guitar Player	Drum Player
Paul	X	X	I
John	X	I	X
Ringo	I	X	X

	Fears		
	13	Apple Computers	Heights
Paul	X	I	X
John	I	X	X
Ringo	X	X	I
`			

	Fears		
	13	Apple Computers	Heights
Bass player	X	X	I
Guitar player	I	X	X
Drum Player	X	I	X

How do we do constraint propagation in this system? Note that we can deduce rules like the following to fill in the three tables:

1. If all but one entry in a row are *X,* then the remaining entry is *I.*
2. If you know *I*(*x,y*) and *X*(*y,z*) then you may conclude *X*(*x,z*).
If two names pick out the same thing (are identical), then they must share all the same properties.
What other rules do we need?
3. If you know *X*(*x,y*) & *X*(*z,y*) then you may conclude *X*(*x,z*)
4. If you know *X*(*x,y*) & *X*(*x,z*) then you may conclude *X*(*y,z*)

Problem 2:Time Travelers' Convention

The MIT Time Travel Society (MITTTS) has invited seven famous historical figures to each give a lecture at the annual MITTTS convention, and you've been asked to create a schedule for them. Unfortunately, there are only four time slots available, and you discover that there are some restrictions on how you can schedule the lectures and keep all the convention attendees happy. For instance, physics students will be disappointed if you schedule Niels Bohr and Isaac Newton to speak during the same time slot, because those students were hoping to attend both of those lectures.

After talking to some students who are planning to attend this year's convention, you determine that they fall into certain groups, each of which wants to be able to see some subset of the time-traveling speakers. (Fortunately, each student identifies with at most one of the groups.) You write down everything you know:

The list of guest lecturers consists of Alan **T**uring, Ada **L**ovelace, Niels **B**ohr, Marie **C**urie, **S**ocrates, **P**ythagoras, and Isaac **N**ewton.

1) **T**uring has to get home early to help win World War II, so he can only be assigned to the 1pm slot.

2) The Course VIII students want to see the physicists: **B**ohr, **C**urie, and **N**ewton.

3) The Course XVIII students want to see the mathematicians: **L**ovelace, **P**ythagoras, and **N**ewton.

4) The members of the Ancient Greece Club wants to see the ancient Greeks: **S**ocrates and **P**ythagoras.

5) The visiting Wellesley students want to see the female speakers: **L**ovelace and **C**urie.

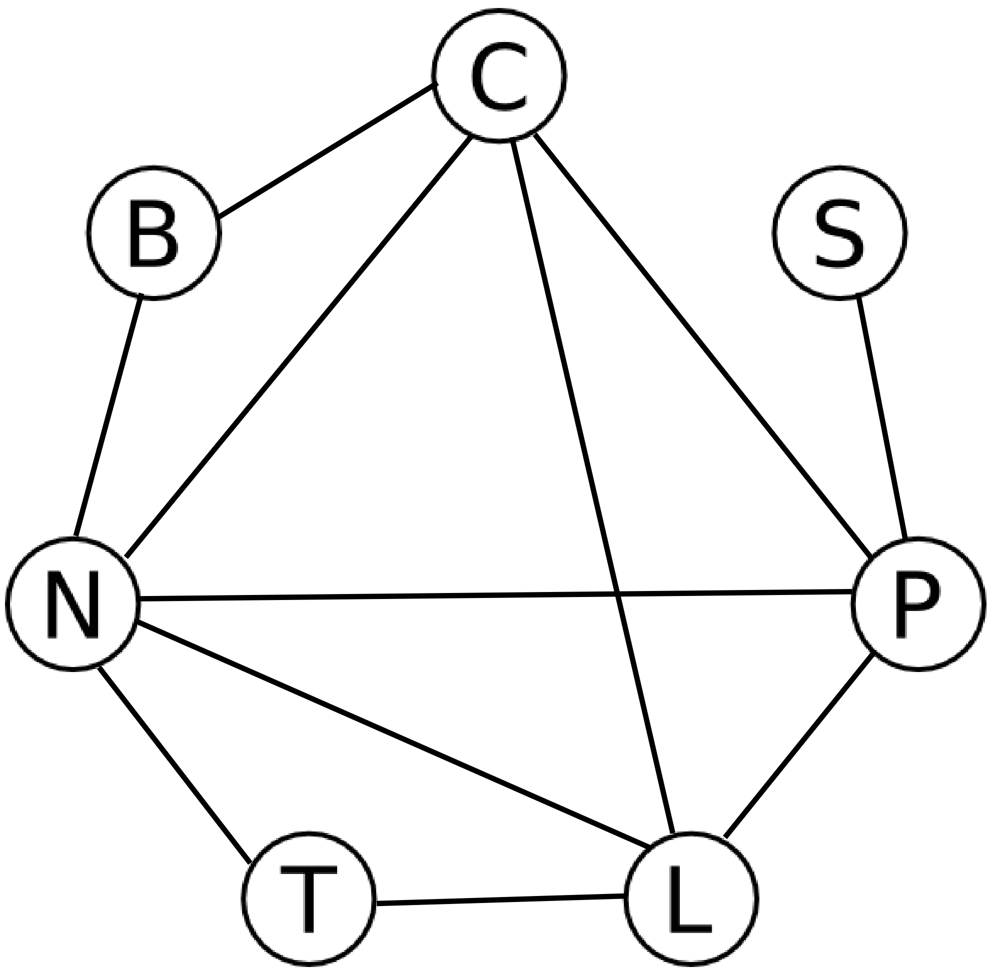
6) The CME students want to see the British speakers: **T**uring, **L**ovelace, and **N**ewton.

7) Finally, you decide that you will be happy if and only if you get to see both **C**urie and **P**ythagoras. (Yes, even if you belong to one or more of the groups above.)

Part A (5 points)

That's a lot of preferences to keep track of, so you decide to draw a diagram to help make sense of it all. Draw a line between the initials of each pair of guests who must not share the same time slot.

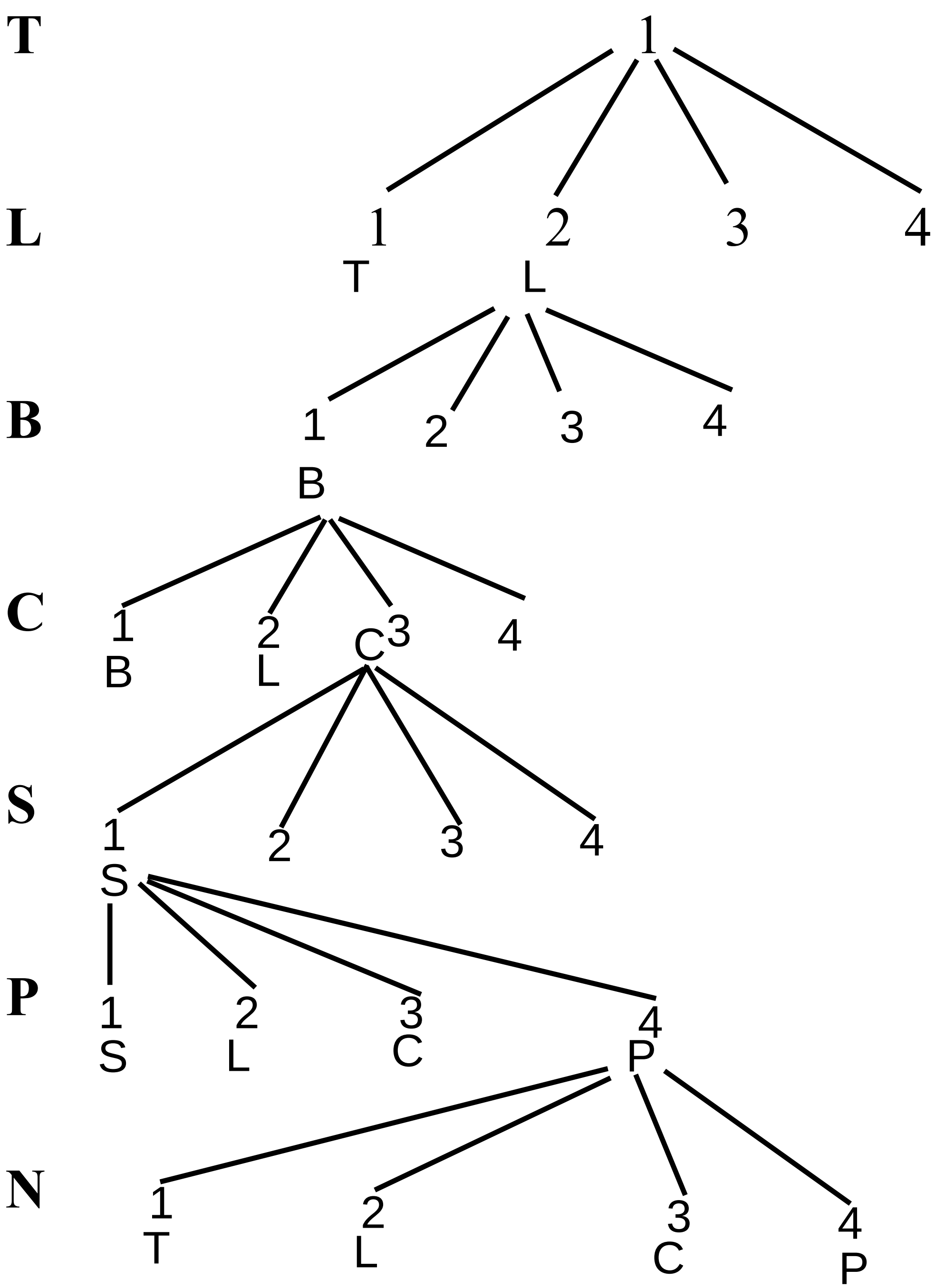
This diagram is repeated on the tear off sheet.



Part B (15 points)

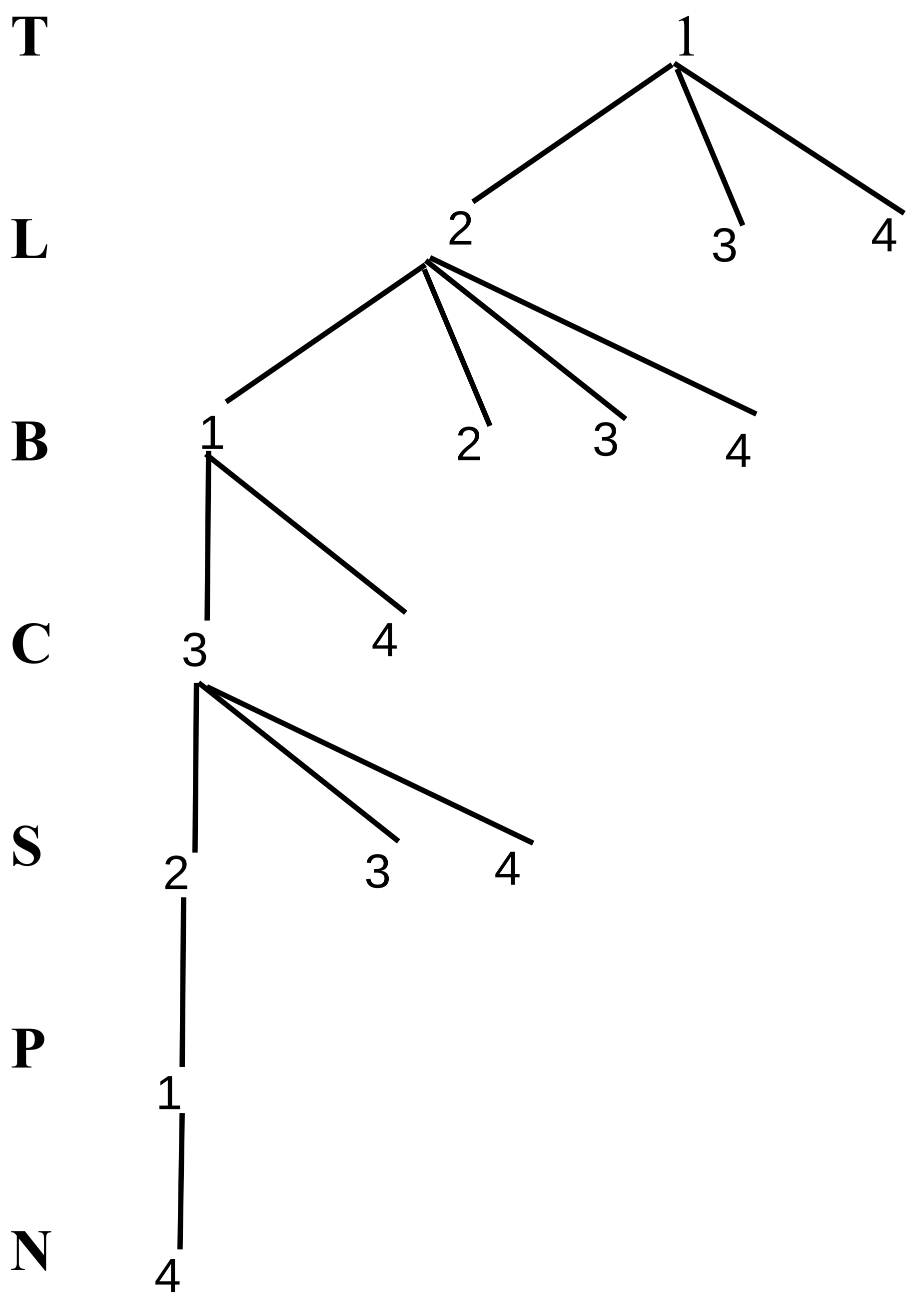
You decide to first assign the time slots (which conveniently happen to be 1, 2, 3, and 4 pm) by using a **depth-first search with no constraint propagation**. The only check is to be sure each new assignment violates no constraint with any previous assignment. As a tiebreaker, assign a lecturer to the earliest available time slot (so as to get them back to their own historical eras as soon as possible).

In the tree below, Alan Turing has already been scheduled to speak at 1 pm, in accordance with constraint #1. Continue filling in the search tree up to the first time you try (and fail) to assign a time slot to Isaac Newton, at which point you give up in frustration and move on to Part C in search of a more sophisticated approach.



Part C (20 points)

You're not fond of backtracking, so rather than wait and see just how much backtracking you'll have to do, you decide to start over and use **depth-first search with forward checking (constraint propagation through domains reduced to size 1)**. As before, your tiebreaker is to assign the earliest available time slot.



What is the final lecture schedule you hand in to MITTTS?

1 pm: T, P, B

2 pm: S, L

3 pm: C

4 pm: N

Part D (10 points)

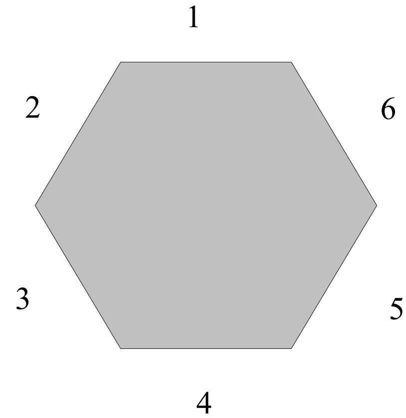
Now, rather than backtracking, you're concerned about the amount of time it takes to keep track of all those domains and propagate constraints through them. You decide that the problem lies in the ordering of the guest list. Just then, you get a call from the MITTTS president, who informs you that **Alan Turing's schedule has opened up and he is now free to speak during any one of the four time slots.**

Armed with this new information, you reorder the guest list to maximize your chances of quickly finding a solution. In particular, which lecturer do you now assign a time slot to first, and why?

Newton - has the most constraints

10/13/11 Constraint Propagation

You just bought a 6-sided table (because it looks like a benzene ring) and want to hold a dinner party. You invite your 4 best friends: McCain, Obama, Biden and Palin. Luckily a moose wanders by and also accepts your invitation. Counting yourself, you have 6 guests for seats labeled 1-6.



Your guests have seven seating demands:

- Palin wants to sit next to McCain
- Biden wants to sit next to Obama
- Neither McCain nor Palin will sit next to Obama or Biden
- Neither Obama nor Biden will sit next to McCain or Palin
- The moose is afraid to sit next to Palin
- No two people can sit in the same seat, and no one can sit in 2 seats.
- McCain insists on sitting in seat 1

Part A (10 points)

You realize there are 2 ways to represent this problem as a constraint problem. For each below, run the domain reduction algorithm and continue to propagate through domains reduced to one value. That is, **cross out all the impossible values in each domain without using any search.**

Variables: You, Moose, McCain, Palin, Obama, Biden
Domains: Seats 1-6
Constraints: I-VII

You:	~~1~~	2	3	4	5	6
Moose:	~~1~~	2	3	4	5	6
McCain:	1	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~
Palin:	~~1~~	2	~~3~~	~~4~~	~~5~~	6
Obama:	~~1~~	~~2~~	3	4	5	~~6~~
Biden:	~~1~~	~~2~~	3	4	5	~~6~~

Variables: Seats 1-6
Domains: You, Moose, McCain, Palin, Obama, Biden
Constraints: I-VII

1:	~~You~~	~~Moose~~	McCain	~~Palin~~	~~Obama~~	~~Biden~~
2:	You	Moose	~~McCain~~	Palin	~~Obama~~	~~Biden~~
3:	You	Moose	~~McCain~~	~~Palin~~	Obama	Biden
4:	You	Moose	~~McCain~~	~~Palin~~	Obama	Biden
5:	You	Moose	~~McCain~~	~~Palin~~	Obama	Biden
6:	You	Moose	~~McCain~~	Palin	~~Obama~~	~~Biden~~

Part B (15 points)

For now, you decide to continue using seats as variables and guests as domains (the 2nd representation). You decide to see how a depth-first search with no constraint propagation works , but as you run the search you see you are doing a lot of backtracking.

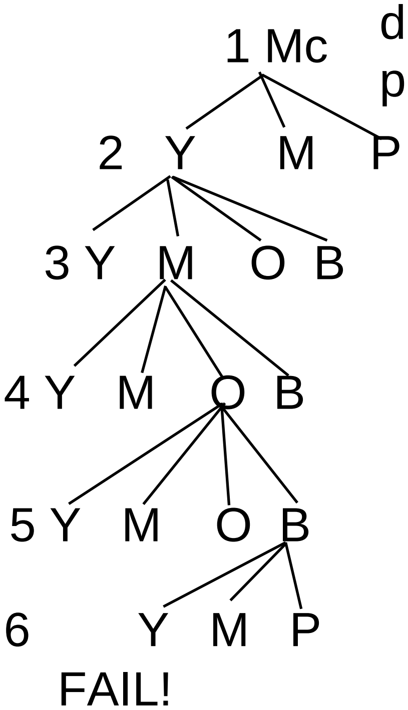
You break ties by choosing the left-most in this order:

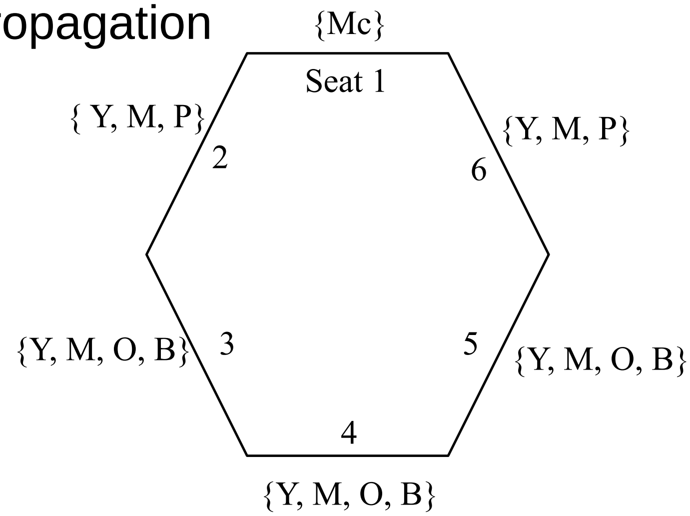
You (Y), Moose (M), McCain(Mc), Palin (P), Obama (O), Biden (B)

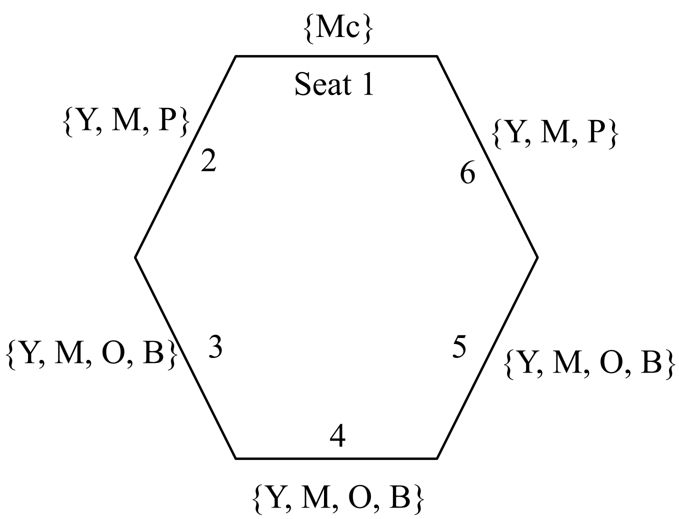
Show the partial search tree *only* up to the first time you try to assign seat 6. Begin with the reduced domains from the previous page. Use only the constraints supplied; use no commensense beyond what you see in the constraints. You might want to work this out on the tear off sheet at the end of the quiz first.

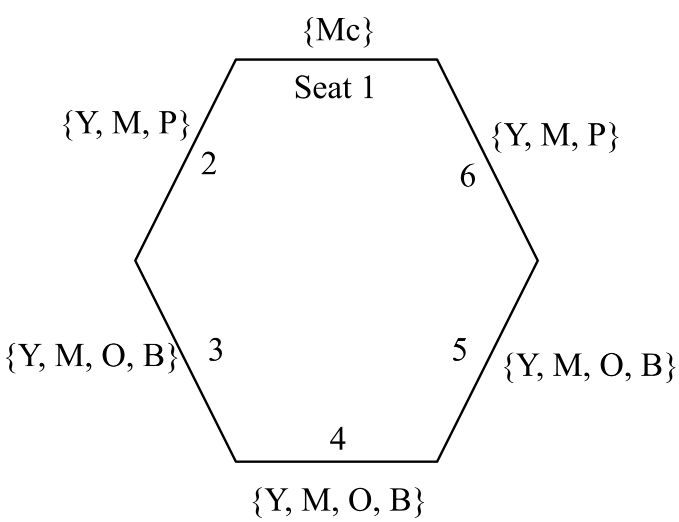
1	Mc
2	Y M P
3	
4	
5	
6	

Depth-first thru already reduced domains - no other constraint propagation









Part C (15 points)

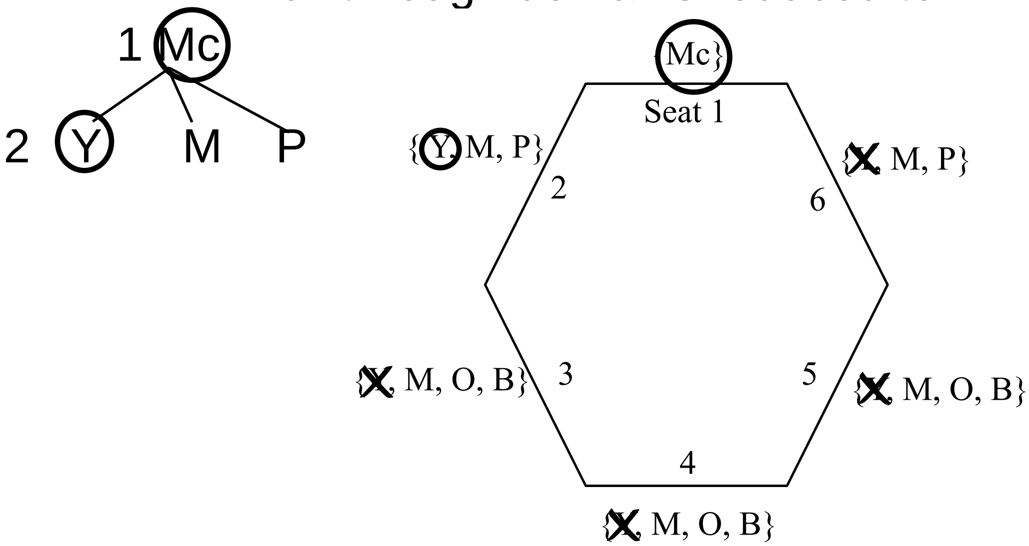
Now you try to use depth-first search *and* ***constraint propagation through domains reduced to size 1.*** You break ties by choosing the left-most in this order:

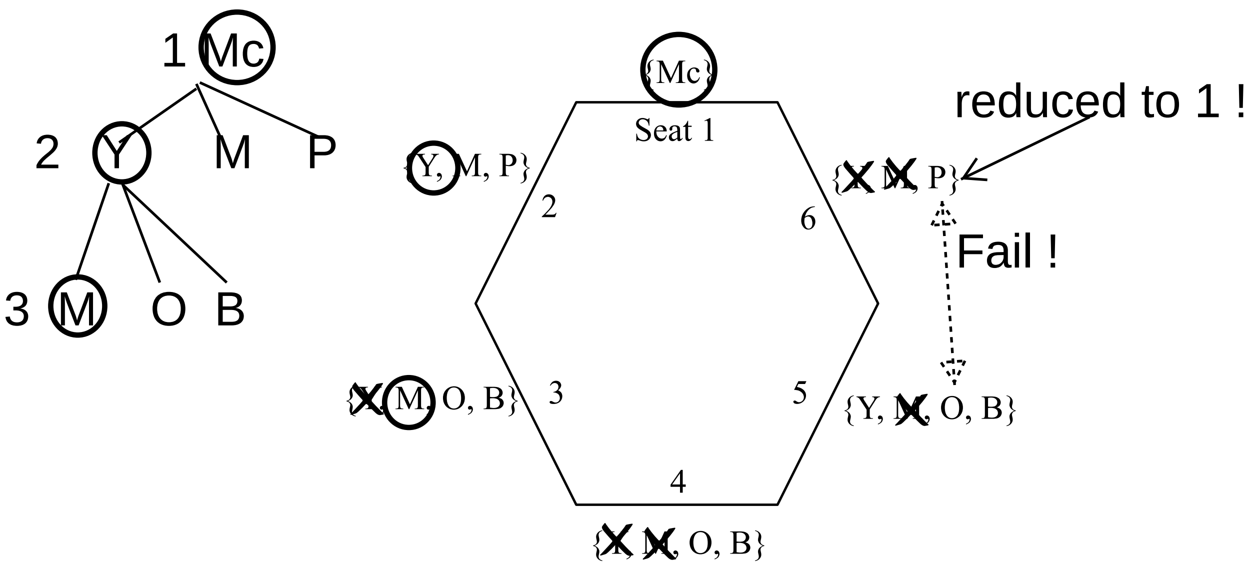
You (Y), Moose (M), McCain(Mc), Palin (P), Obama (O), Biden (B)

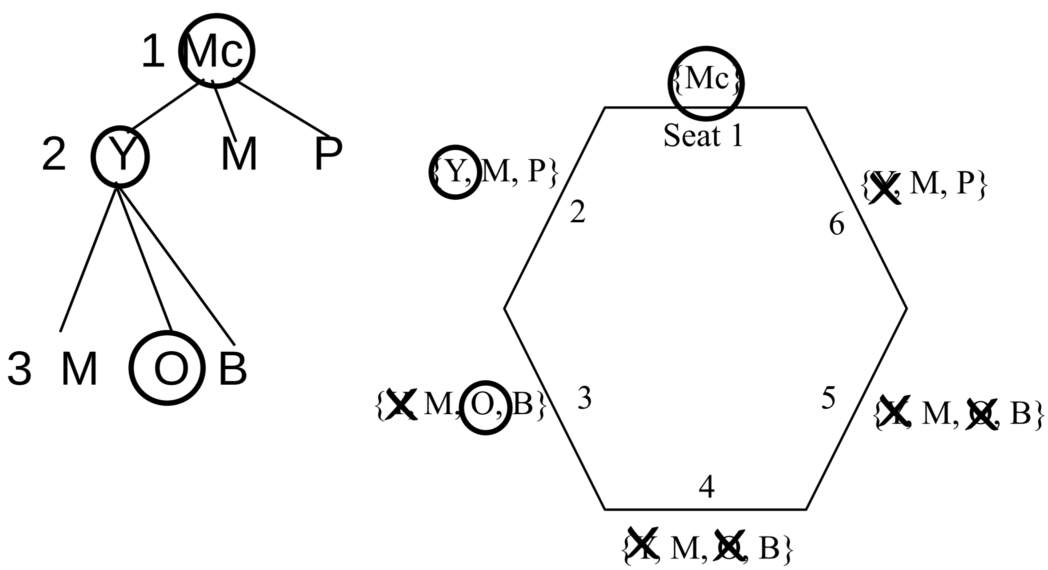
Show the the *full* search tree (up until you find a solution), starting with the same pre-reduced domains. You might want to work this out on a copy at the end of the quiz..

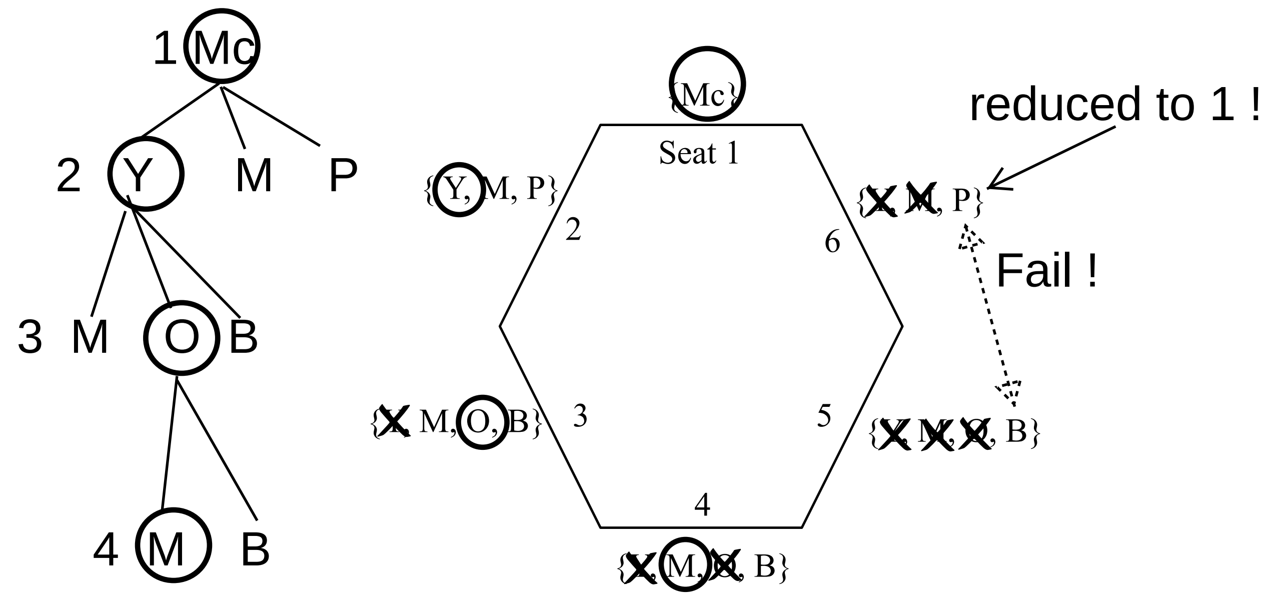
1	Mc
2	Y M P
3	
4	
5	
6	

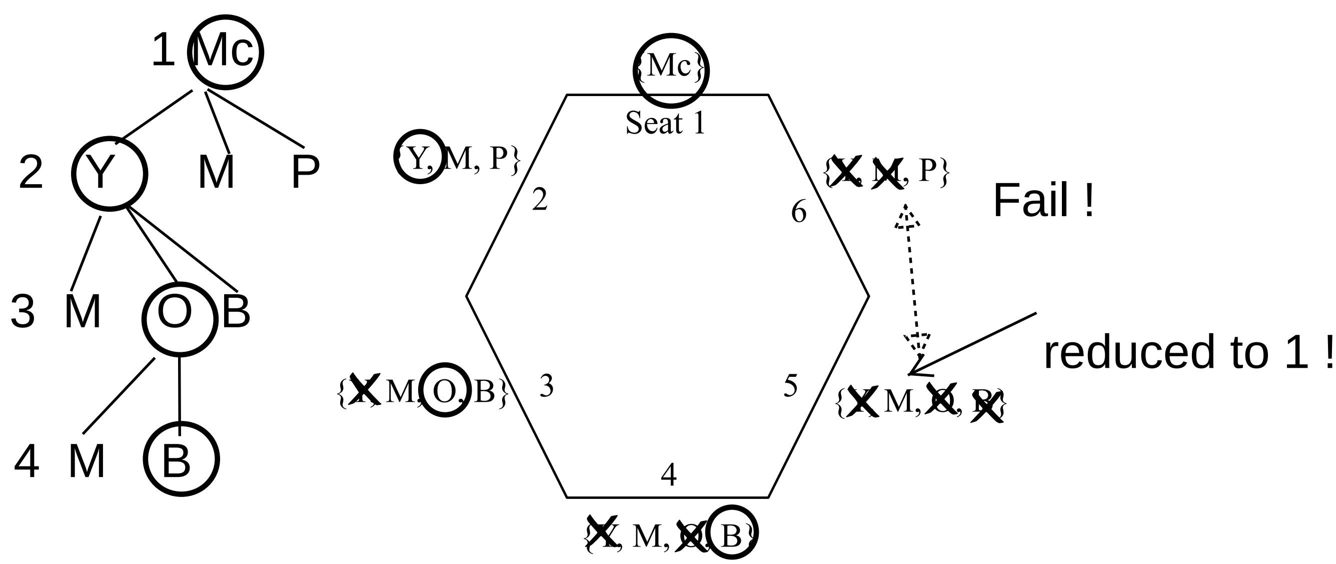
Depth-first through already reduced domains <u>PLUS</u> constraint propagation incl. through domains reduced to 1

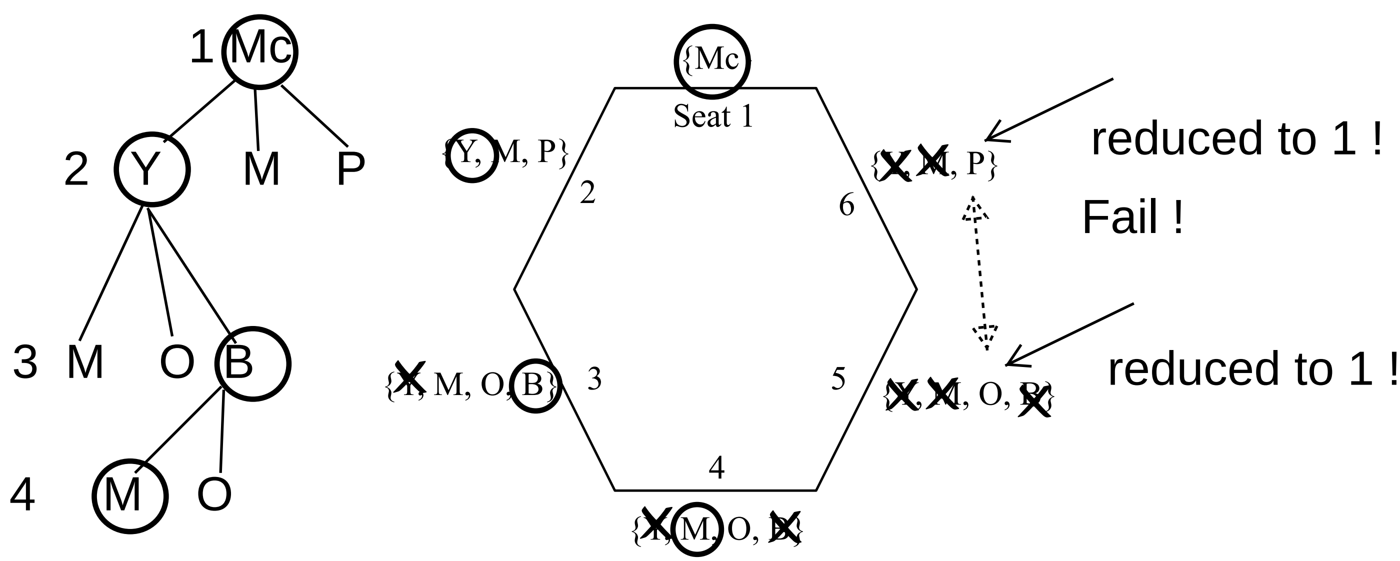




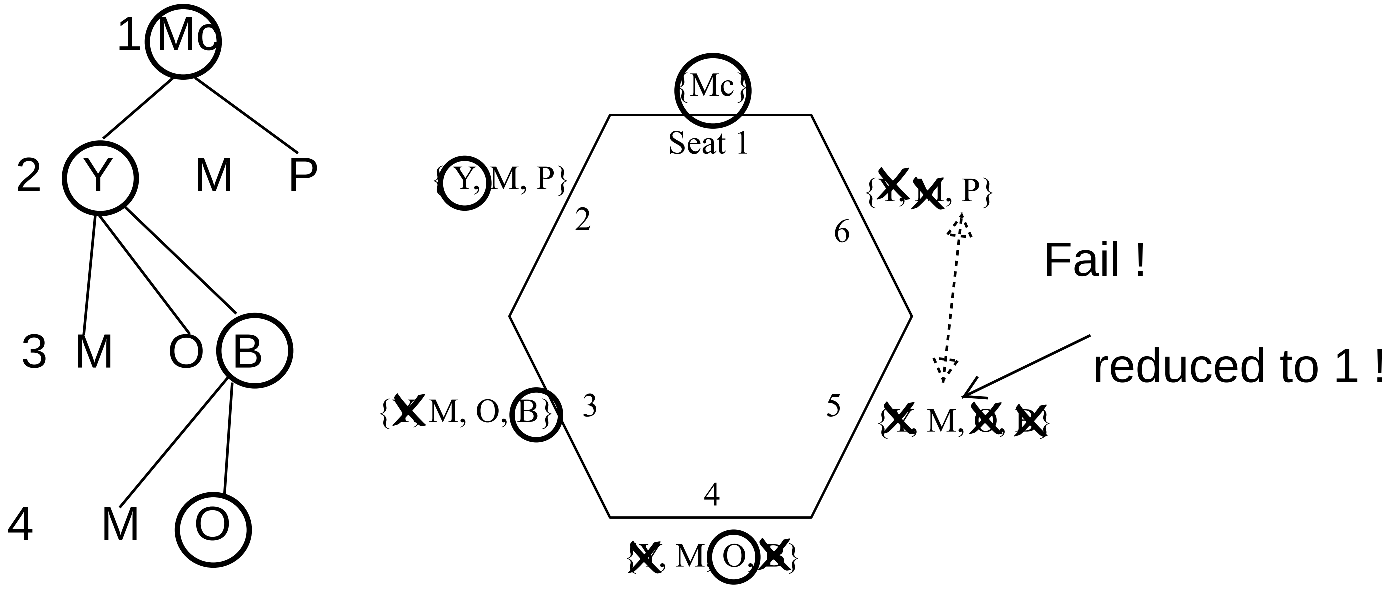


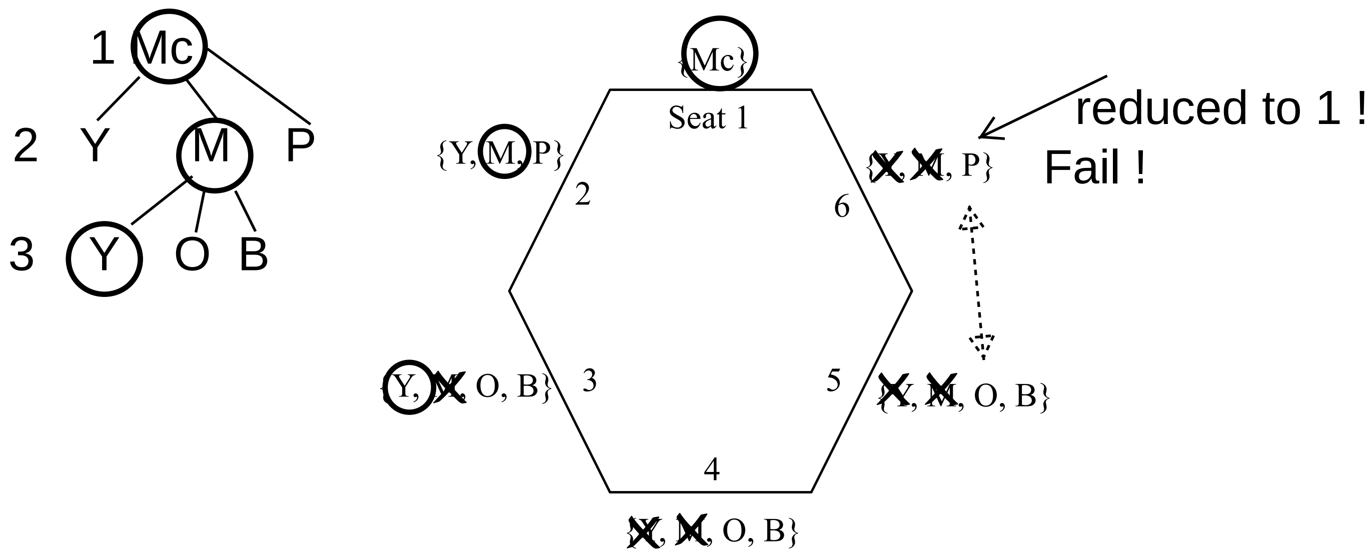


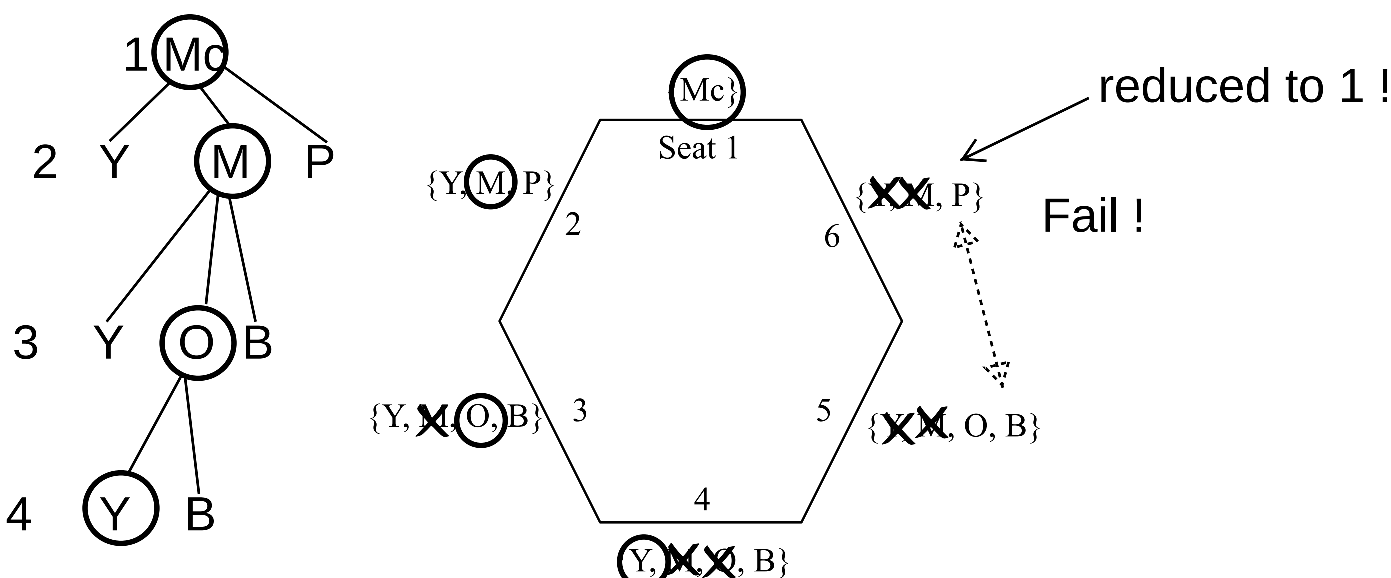




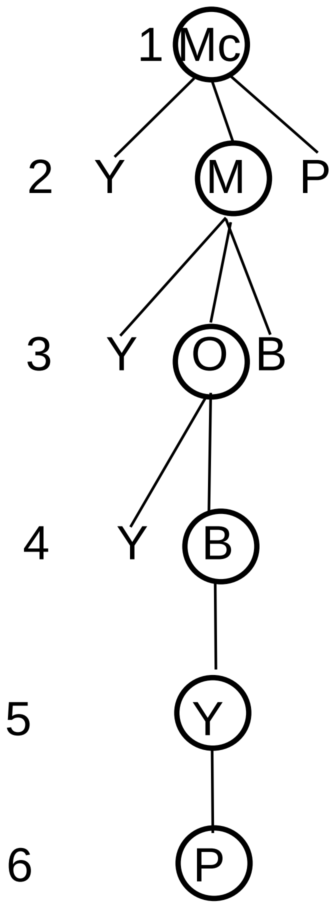
Note: if we enforce constraint that Biden must be next to Obama, then M is ruled out immediately at this point.

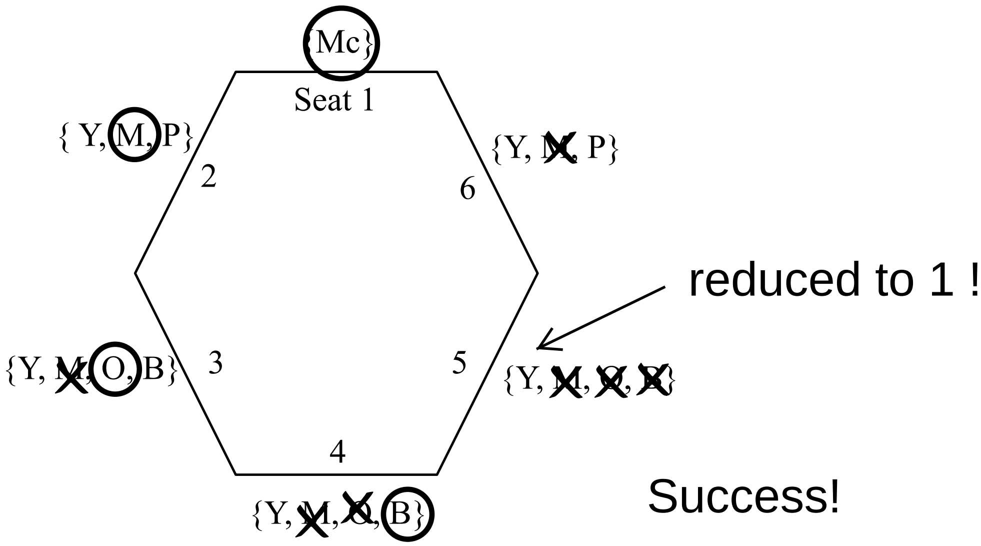


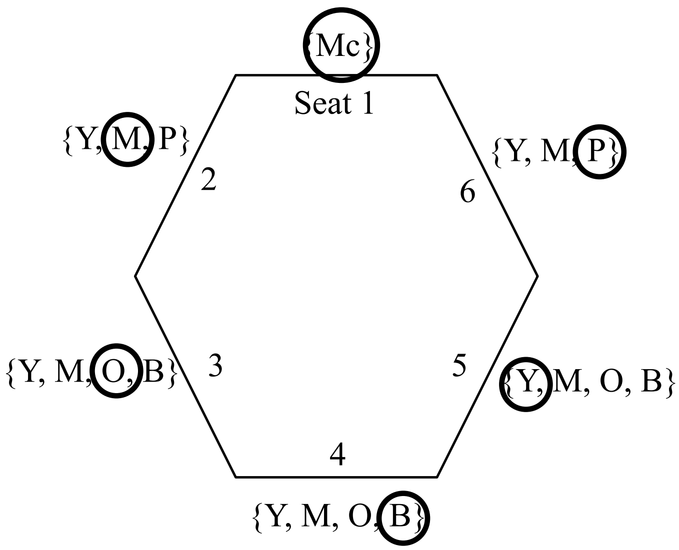


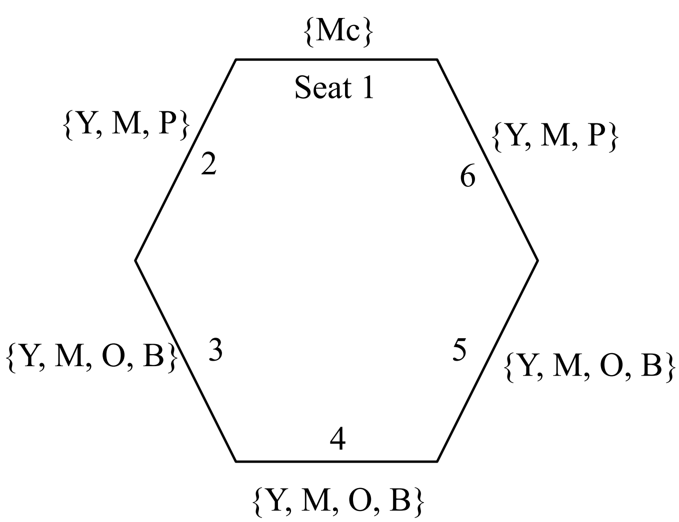


Note: if we enforce constraint that Biden must be next to Obama, then Y is ruled out immediately at this point.


1 Mc
2 Y M P
3 Y O B
4 Y B
5 Y
6 P


{Mc}
Seat 1
{ Y, M, P}
2
{Y, M, P}
6
{Y, M, O, B}
3
5
{Y, M, O, B}
4
{Y, M, O, B}
reduced to 1 !
Success!


{Mc}
Seat 1
{Y, M, P}
2
{Y, M, P}
6
{Y, M, O, B}
3
5
{Y, M, O, B}
4
{Y, M, O, B}


{Mc}
Seat 1
{Y, M, P}
2
{Y, M, P}
6
{Y, M, O, B}
3
5
{Y, M, O, B}
4
{Y, M, O, B}