Rec 2.
9-13-2011

Recitation 2

Last Time

① Information $\boxed{I = \log_2\left(\frac{1}{p}\right)}$ bits ~ measure of uncertainty

— More info revealed by less probable event outcome!

(eg1) Information conveyed or revealed by announcing independent events $A \sim p(A)$ and $B \sim p(B)$

$$I_{AB} = \log_2\left(\frac{1}{P_A \cdot P_B}\right) = \log_2\left(\frac{1}{P_A}\right) + \log_2\left(\frac{1}{P_B}\right)$$

$$= I_A + I_B$$ ~ information is additive! over independent events

② Entropy

~ expected (or average) information (uncertainty) over all possible outcomes.

$$\boxed{H(X) = E[I(X)] = \sum_{N} p(x) \log_2\left(\frac{1}{p(x)}\right)} \text{ bits.}$$

Case 1: ~~An~~ certain event!

Contributes '0' bits to entropy! $\sim (1)\log_2(1) = 0$

Case 2: Likewise, uncertain (or impossible) event

Contributes $(0)\log_2\left(\frac{1}{0}\right)$ → Need L'Hospital's rule!

~~$p(x)$~~ . . $= $ '0' bits

Another Example: (from class notes)

Ex. 2 X is an unknown 4-bit binary number picked @ random.

Y is another 4-bit number given that Hamming dist. btn X and Y is 2.

Q: How many bits of info have you been given?

Soln: There are 2^4 possibilities of all 4-bit sequences $= 16$.

" " ${}^4C_2$ possibilities where 2-positions differ $= \frac{4!}{2!\,2!} = 6$

Thus $p = \frac{6}{16}$

and $I = \log_2\left(\frac{1}{(6/16)}\right) = \log_2(2.667) = \underline{1.42 \text{ bits}}$.

3. Compression:

– Use fewest num. bits to transmit messages!
(BW, P, other considerations).

Trick Idea: Use 'smaller' codes for frequently occuring symbols.
'longer' " " rarely " "

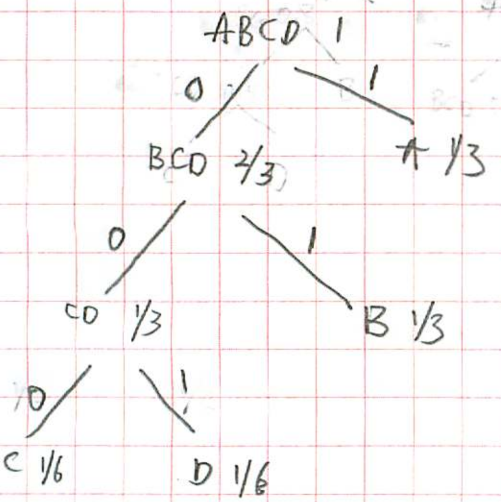
Most frequently occuring alphabets ~ t, e, a, ...
Rarest " " ~ z, Q, ...

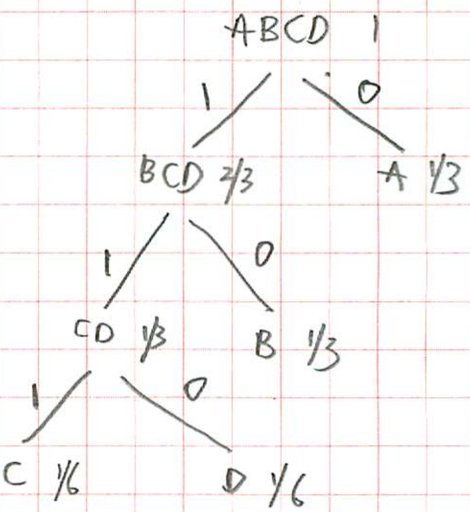
Huffman codes:

Property 1: Non-uniqueness

* 0/1 labels on any pair of branches in a code tree can be reversed

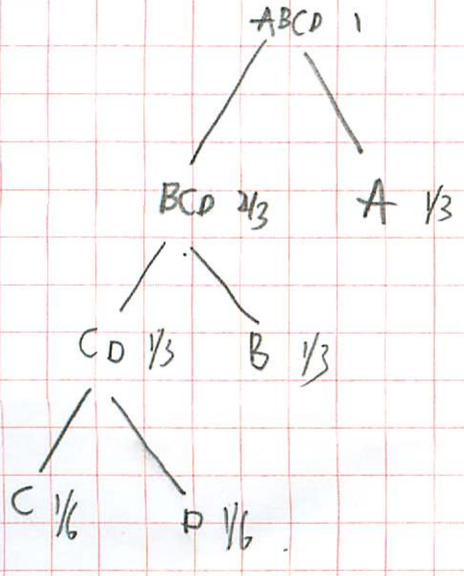
Ex1: A: 1/3 B: 1/3 C: 1/6 D: 1/6.

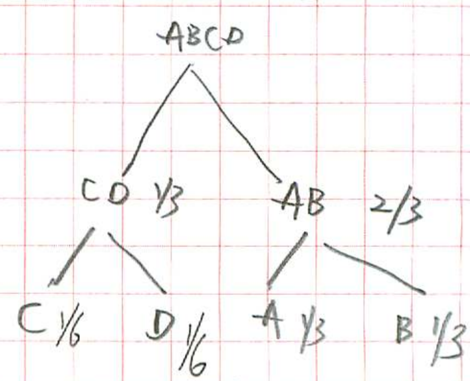




* Huffman coding can produce different non-isomorphic code trees.

Ex: A: 1/3 B: 1/3 C: 1/6 D: 1/6.





$\boxed{\text{Property 2:}}$ Huffman coding over a set of symbols with known probabilities produces a code tree whose expected length is optimal.

Huffman coding with grouped symbols.

* To group symbols into large "metasymbols" and encode those metasymbols instead
* get some gain in compression but at a cost of encoding and decoding complexity

Ex: send messages containing 1000 6.02 grades of A, B, C, D.
A: 1/3 B: 1/2 C: 1/12 D: 1/12

Size of grouping	Expected length for 1000 grades
1	1667
2	1646
3	1637
4	1633

Another Example:

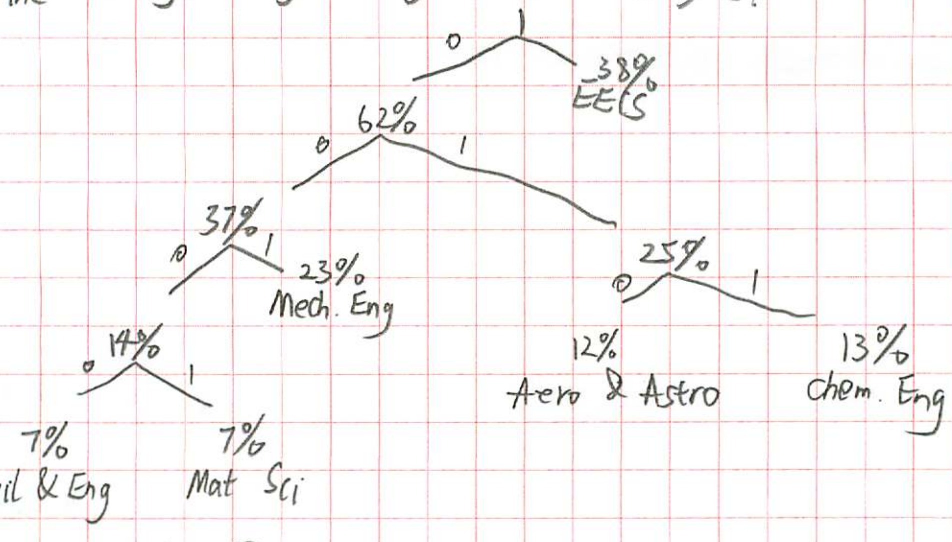
The following table shows the student enrollements for the school of Engineering.

Civil & Env	7%
Mech. Eng.	23%
Mat Sci	7%
EECS	38%
Chem. Eng.	13%
Aero & Astro	12%

Q1: design a Huffman code

Q2: If you code is used to send messages containing only the encoding of the departments for each students in group of 100 randomly chosen students, what is the average length of such messages?

A1:



Civil & Eng	0000
Mech. Eng.	001
Mat. Sci.	0001
EECS	1
Chem. Eng.	011
Aero & Astro	010

A2: average message length $= 100 * \sum P_i *$ (length of code)

$= 100 * (7\% * 4 + 23\% * 3 + 7\% * 4 + 38\% * 1 + 13\% * 3 + 12\% * 3)$

$= 238$