Fractional & Integral Knapsack have optimal substructure.

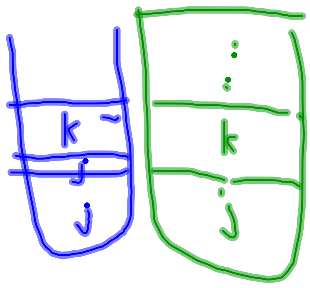
Only fractional problem has greedy choice property.

Statement

Let j be item w/ maximum $\frac{v_i}{w_i}$. Then $\exists$ an opt. sol'n in which you take as much of j as possible.

Pf Suppose you didn't take as much as possible of item j, (and the knapsack is full). Then there exists some item k, w/ $\frac{v_k}{w_k} < \frac{v_j}{w_j}$ that is in the knapsack.

Oct 20-4:08 PM



not all of j is in

Then take a wt. ε piece of k out, add a wt. ε piece of j in.

Increase knapsack value by

$$\varepsilon \frac{v_j}{w_j} - \varepsilon \frac{v_k}{w_k} = \varepsilon\left(\frac{v_j}{w_j} - \frac{v_k}{w_k}\right) > 0 \quad \otimes$$

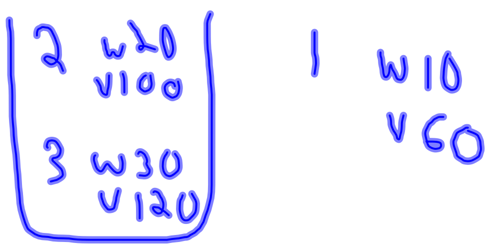
Oct 20-4:17 PM

<u>Alg</u>

1. Sort items by v_j / w_j, renumber.
2. For i = 1 to n

   Add as much as possible of item i

Oct 20-4:23 PM

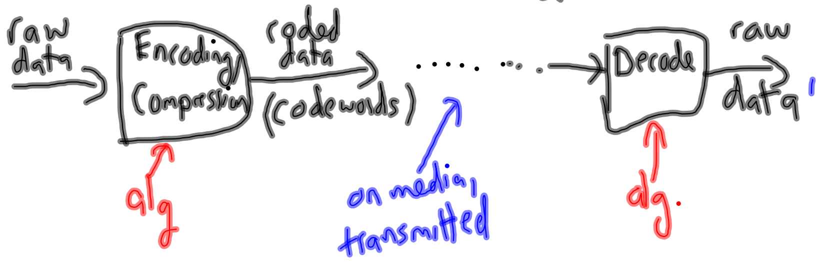


Oct 20-4:28 PM

Huffman Codes

(compression)

- data on media (CD, DVD)
- data over the internet



Compression

→ lossless - encode/decode → get back original data

lossy - " → get back an approx. of original data

Oct 20-4:29 PM

Encode English letters

Standard encoding

ASCII - 8 bits

$2^8 = 255$

standard encoding of k symbols, you need $\lceil \lg k \rceil$ bits

Oct 20-4:35 PM

encoding alg.

face

10100001010100

decoding alg.

"code"

Oct 20-4:39 PM

Take advantage of different frequences of letters in English.

E T A S I O ... .

Oct 20-4:44 PM

huff.pdf (application/pdf Object) - Mozilla Firefox

http://www.columbia.edu/~cs2035/courses/csor4231.F09/huff.pdf

Different types of codes

- **fixed length code.** Each codeword uses the same number of bits.
- **variable length code.** Codewords can use differing numbers of bits.

Example

character	frequency	fixed length code	variable length code
a	.45		1
b	.13		1
c	.12		2
d	.16		2
e	.09		2
f	.05		2

$$.45(1)+.13(1)+.12(2)+.16(2)+.09(2)+.05(2) < 2$$

Oct 20-4:48 PM

a	0
b	1
c	00
d	01
e	10
f	11

face

1100010

bbaaaba

fcda

⋮

Oct 20-4:50 PM

huff.pdf (application/pdf Object) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.columbia.edu/~cs2035/courses/csor4231.F09/huff.pdf

- variable length code. Codewords can use differing numbers of bits.

character	frequency	fixed length code	variable length code
a	.45	000	0
b	.13	001	101
c	.12	010	100
d	.16	011	111
e	.09	100	1101
f	.05	101	1100

1100 0 100 1101
f a c e

Evaluation of code: Expected number of bits per codeword.

Fixed length code: 3

Variable length code:

$$.45(1) + .13(3) + .12(3) + .16(3) + .09(4) + .05(4) = 2.24$$

Prefix free codes: No codeword is a prefix of any other codeword.

11.00 x 8.50 in

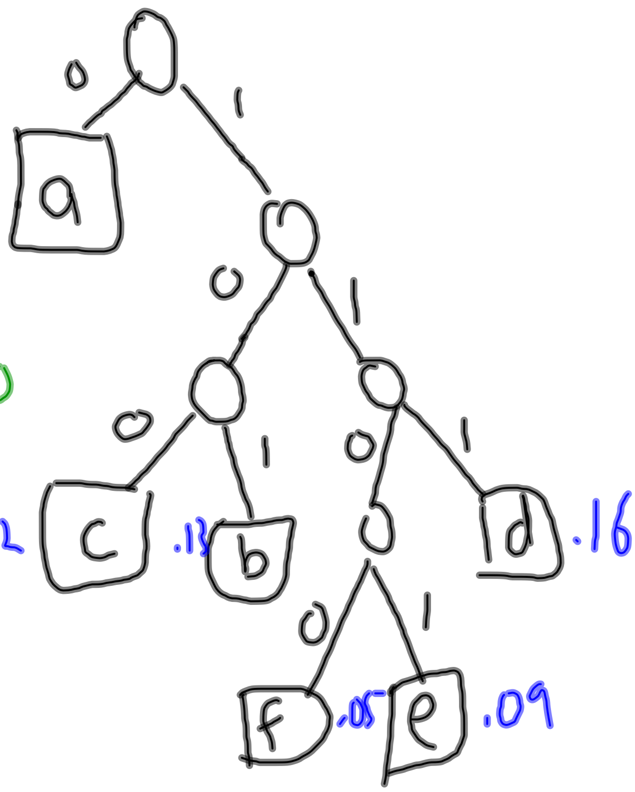
Oct 20-4:54 PM

a	0	0
b	101	10
c	100	110
d	111	1110
e	1101	11110
f	1100	111110

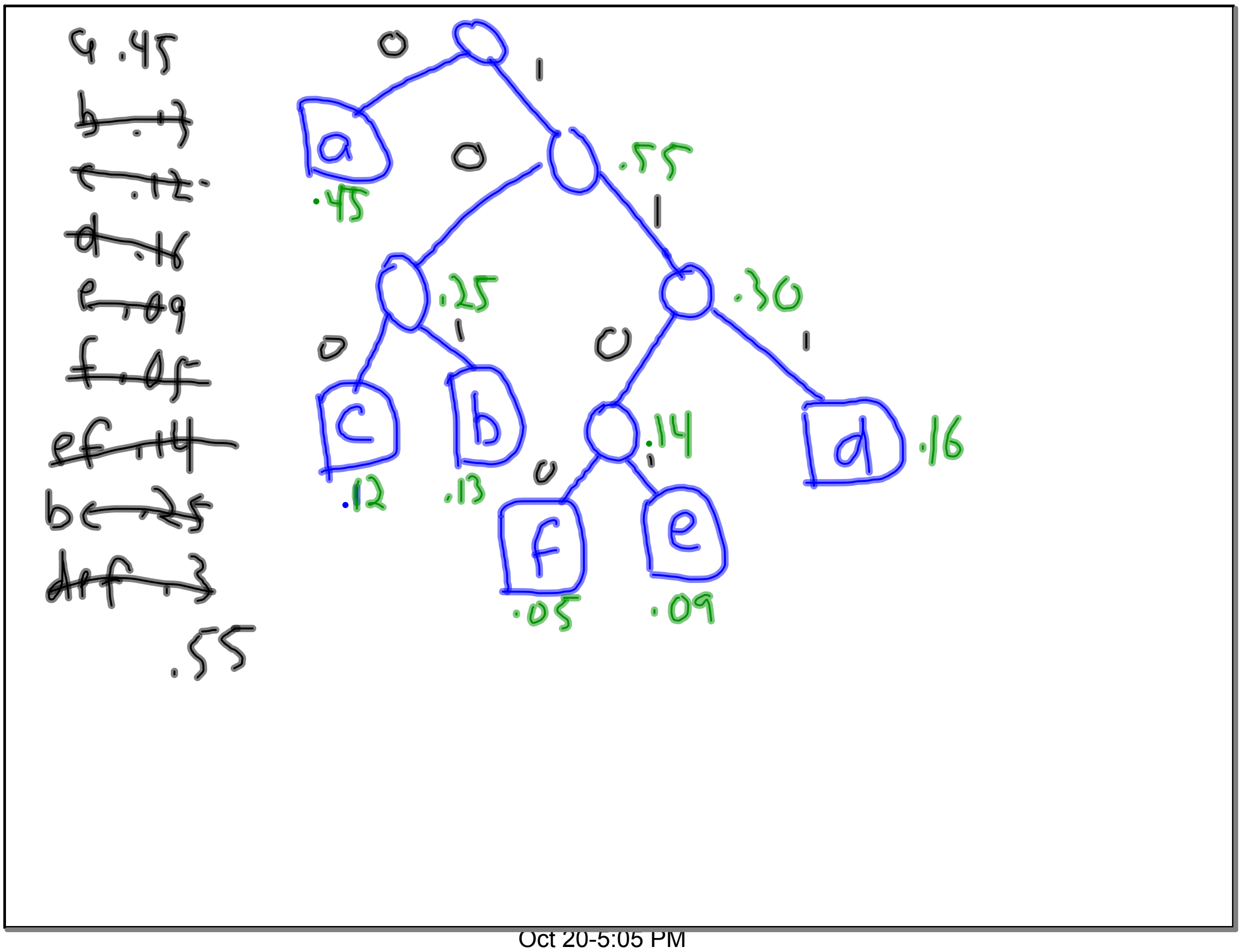
path ⟷ codewords

face

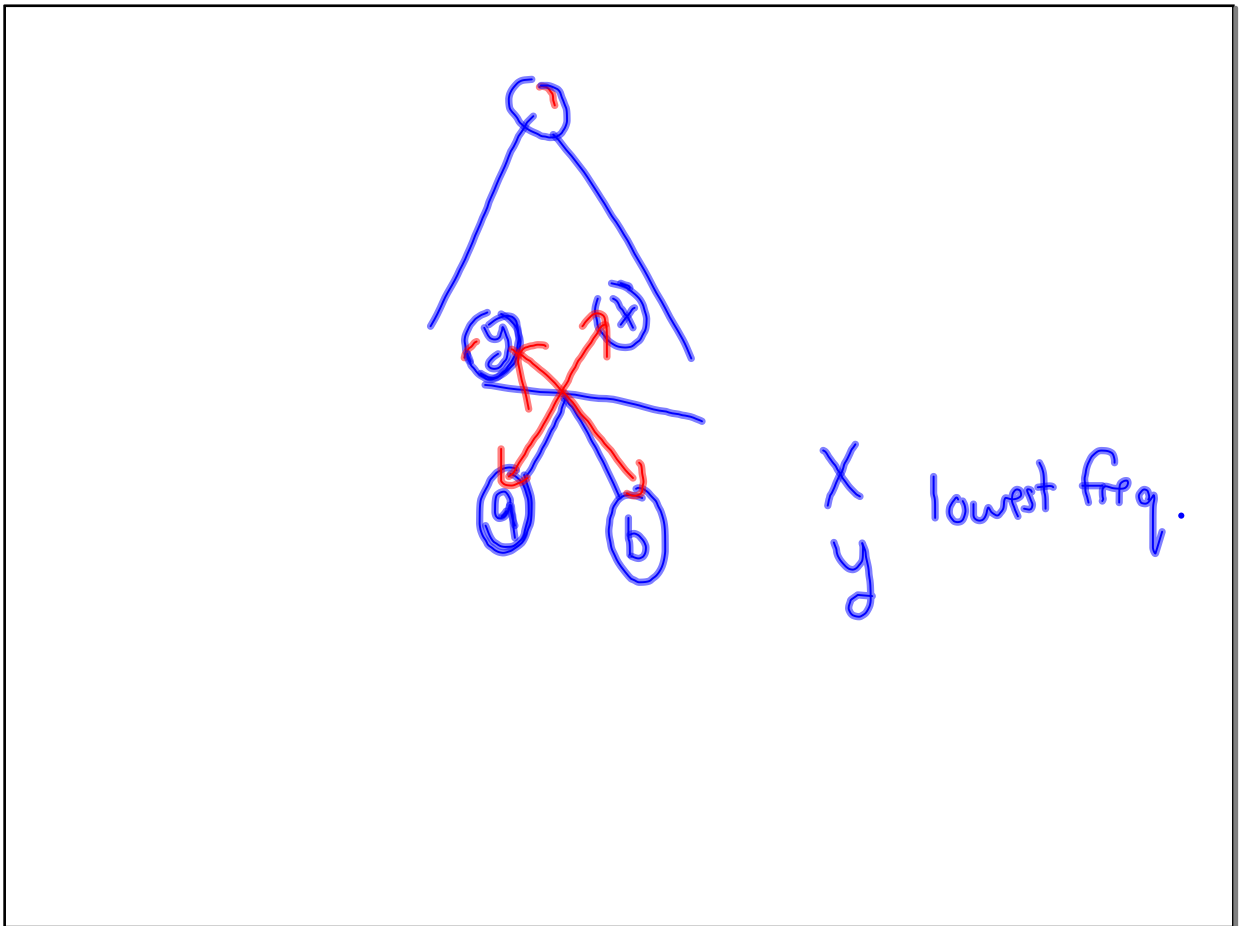
1100 0 100 1101
f a c e



Oct 20-4:54 PM



Oct 20-5:05 PM



Oct 20-5:13 PM

huff.pdf (application/pdf Object) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.columbia.edu/~cs2035/courses/csor4231.F09/huff.pdf

- Let a and b be two characters that are sibling leaves of maximum depth in T. (wlog, $f[a] \leq f[b]$ and $f[x] \leq f[y]$.)
- $f[x] \leq f[a]$ and $f[y] \leq f[b]$, since $f[x]$ and $f[y]$ are the two lowest leaf frequencies.
- Exchange the positions in T of a and x to produce a tree T'.
- Exchange the positions in T' of b and y to produce a tree T''.

Now look at the difference between $B(T)$ and $B(T')$

$$\begin{aligned}
B(T) - B(T') &= \sum_{c \in C} f(c) d_T(c) - \sum_{c \in C} f(c) d_{T'}(c) \\
&= f[x] d_T(x) + f[a] d_T(a) - f[x] d_{T'}(x) - f[a] d_{T'}(a) \\
&= f[x] d_T(x) + f[a] d_T(a) - f[x] d_T(a) - f[a] d_T(x) \\
&= (f[a] - f[x])(d_T(a) - d_T(x)) \\
&\geq 0 ,
\end{aligned}$$

Reasons for last inequality:

11.00 x 8.50 in

Oct 20-5:19 PM

Assuming
- each character is encoded individually into an integral # of bits,

Huffman coding is optimal (minimized expected # of bits transmitted)

Oct 20-5:16 PM

Oct 20-5:24 PM