Knapsack Problem

Objects (o)	1	2	3	4	5	6	7
Profit (P)	10	5	15	7	6	18	3
weight (w)	2	3	5	7	1	4	1

n = 7

m = 15 ⇒ bag ⇒ Knapsack

<u>Size</u>

Knapsack = Container loading problem

bag capacity = 15 kg ⇒ could be any thing

fill bag with objects with max profit

⇒ loading Container etc

⇒ Trade etc

⇒ Optimization problem / maximization

Problem = max profit

⇒ Objective max {$x_i p_i$}

⇒ Constraint

max 15 kg ⇒ size of knapsack

$\sum x_i w_i \leq m$

How to Solve

x (   )

↗ x_1 x_2 x_3 x_4 x_5 x_6 x_7

To keep track if item is included

$0 \leq x \leq 1$ ⇒ we can take fraction

e.g 7kg potato, we take 2 kg.

We will select highest profit/weight.

So

obj	1	2	3	4	5	6	7
profit	10	5	15	7	6	18	3
weight	2	3	5	7	1	4	1
P/w	5	1.3	3	1	6	4.5	3

Now use greedy approach

(3)

$X (x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7)$

1

Now $15 - 1 = 14$

$X (x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7)$

1 1

1 1

1

2/3

0

Total ~~profit~~ weight

$\Sigma x_i w_i$

$1 \times 2 + \frac{2}{3} \times 3 + 1 \times 5 + 0 \times 7 + 1 \times 1 + 1 \times 4 + 1 \times 1$

$= 15$ kg.

Total profit

$\Sigma x_i p_i \quad 1 \times 10 + \frac{2}{3} \times 5 + 1 \times 15 + 0 \times 7 + 1 \times 6 + 1 \times 18$
$+ 1 \times 3 = 54.6$

0-1 Knapsack

$m = 8$ $\quad P = \{1, 2, 5, 6\}$

$n = 4$

$w = \{2, 3, 4, 5\}$

object $\{1, 2, 3, 4\}$

P	w_i	W: 0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	1	1	1	1	1	1	1
2	3	0	0	1	2	2	3	3	3	3
5	4	0	0	1	2	5	5	6	7	7
6	5	0	0	1	2	5	6	6	7	8

$V\{i, w\} = \max\{V\{i-1, w\}, V\{[i-1, w - w_i] + P[i]\}\}$

$V[i, w] = \max\{V[i-1, w],$

$\cancel{V\{i-1, w[i]}$ weight

$V[i-1, w - w[i]] + P[i]\}$

$V[4, 1] = \max\{V[4-1, 1],$ ~~V[~~

$V[4-1, 1-5+6]$

5

$V[4,5] = \max\{V[3,5], V[3, 5-5] + 6\}$

$= \max[V[3,6], V[3,6]]$

x_1 x_2 x_3 x_4

0 1 0 1

Check max profit in last 4

it is 8.

So add 4.

So

8 - profit of 4

= 2.

Check 2nd last row, if

it has two, then yes ~~take it~~. Check

it exist in previous, then take it.

n log n

0-1 Knapsack

Obj 1 2 3

w 2 4 8

⇒ 20 25 60

10 6.5

Greedy will not a

Huffman Code

→ Compression technique

→ Greedy Algo

message = BCCA BBDDAE CC

BBAEDDCC

cost of message = 20 ~~bits~~ byte = Length

160 bits

A 65 01000001

Make your own encoding 7

eg. ~~1~~ 1 bit 0/1 = represent

2 ~~value~~ bit represent 4 value

~~4 bit~~ 3 bit = 8 value

0 - - - 7

So

A	~~5~~	3	000
B	~~8~~	~~6~~ 5	001
C		6	010
D		4	011
E		2	100

Size of encoding = 5 x 8bit 5x3

40 + 15

= 55 bit

total = 55 + 60 = 115 bits

Steps Huffman

1- find ~~pe~~ frequency of all alphabet

A = 3

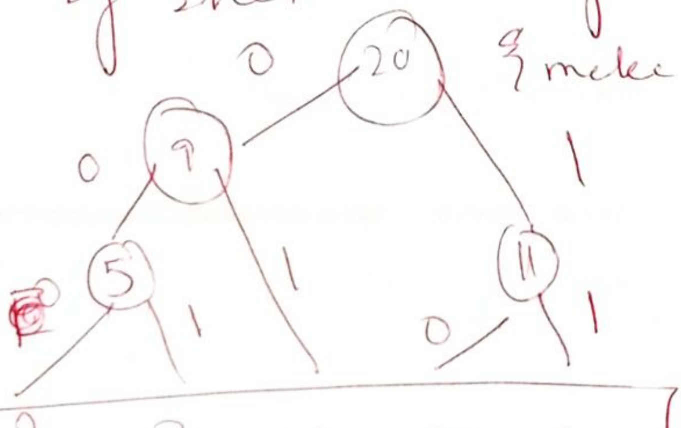
B = 5

C = 6

D = 4

E = 2.

2- Sort by Shel. 3- merge two smallest & make tree



2	3	4	5	6
E	A	D	B	C

A = 001
B = 10
C = 11
D = 01
E 000
↓
45 bits for bin

All left zero, all right 1

Optimal merge pattern tree

~~(8×5 = 40 + 45) = 95~~

5×40+12 = 52 bit for table

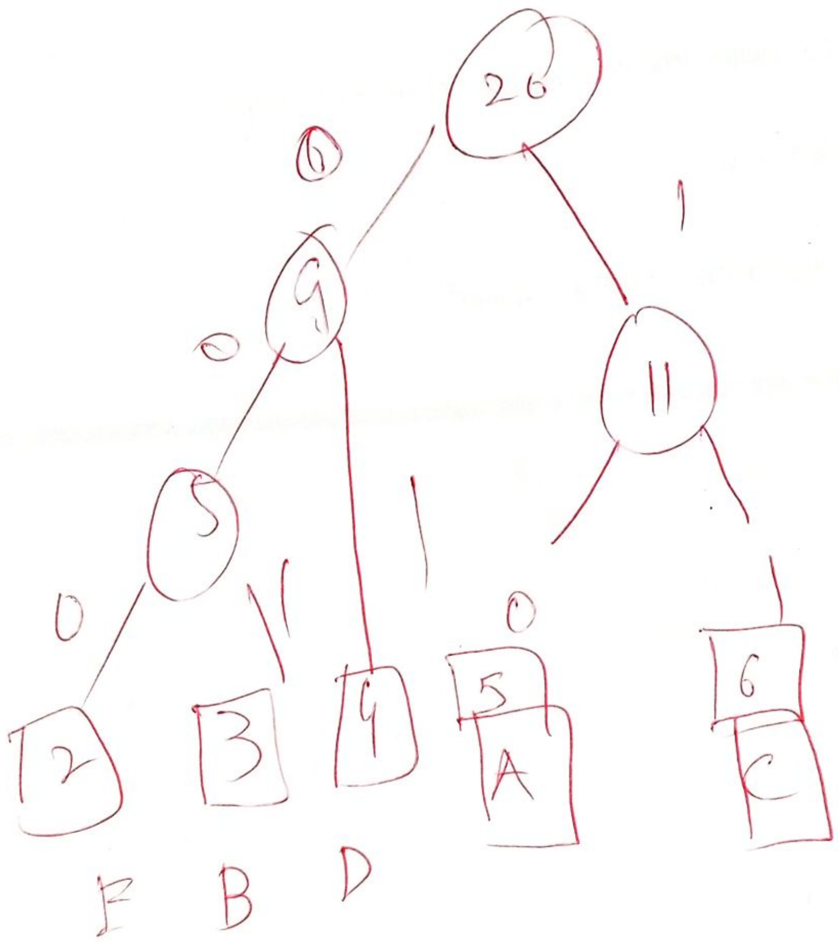
45+12 = 97

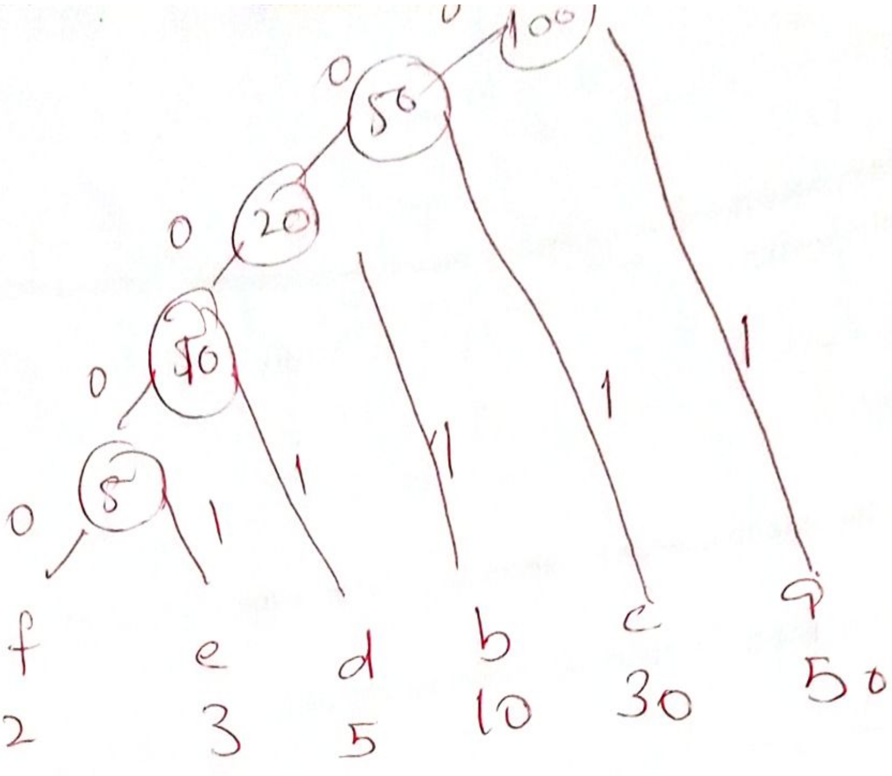
To decode

Start from tree

message = B C C D A C

001 11 11 01 10 11





a = 1 b = 001 c = 01

d = 0001 e = 00001

f = 00000

= 185 bits