NAME:- KRUNAL RANK

ADM. NO:- U18C0081

B TECH 3RD YEAR

U18CO081 15/10/20

DAA
Tutorial 7

Ans 1: For the given problem, let us use dynamic programming. Here dp[i][j] denotes the max. value obtained using items 1 to i with permissible weight of knapsack as j.

Then, the table looks like:-

dp	1	2	3	4	5	6	7
1	0	3	3	3	3	3	3
2	0	3	3	3	6	6	9
3	0	3	4	4	7	7	9
4	0	3	4	5	7	7	9

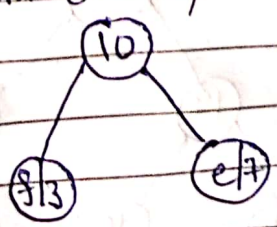
Hence, answer is 9, that includes item (2,3) and (5,6)

Ans 2: First let us make a minheap using given characters and its frequencies.

Hence,

f	3
e	7
c	9
b	11
d	14
a	48

Now, pop two characters with minimum frequencies and make a tree with their sum of frequencies as root. Hence,

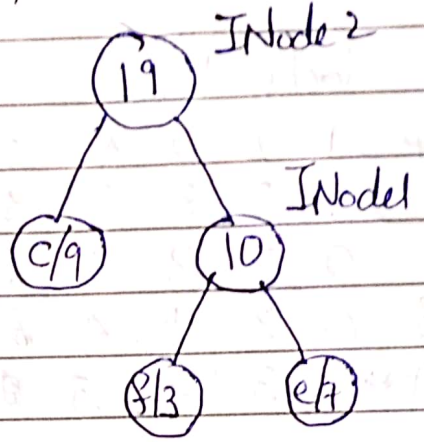


Push it into minheap.

Minheap now:-

c	9
Inode 1	10
b	11
d	14
a	48

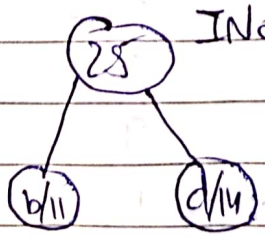
Repeat the same process.



Minheap:-

~~INode2~~	~~19~~
b	11
d	14
INode2	19
a	48

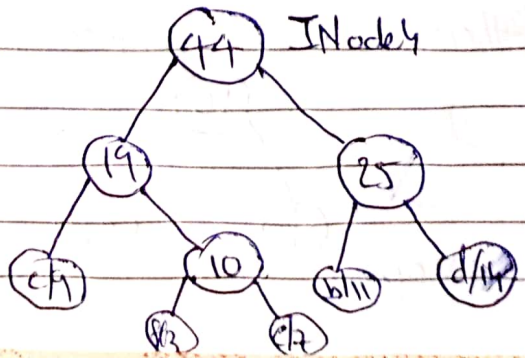
Then,



Minheap:-

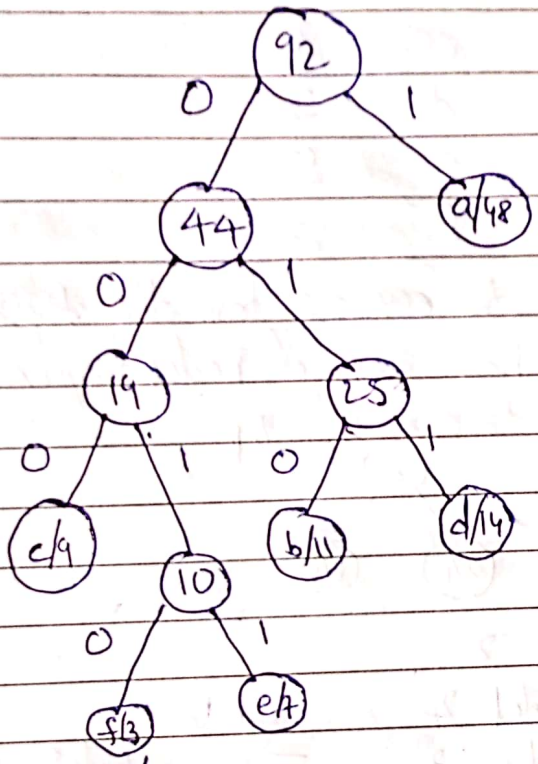
INode2	19
INode1	25
~~In~~ a	48

Then,



Then, Minheap :- INode4 44
a 48

Hence, final Huffmann Tree :-



Also Huffmann code :-

a	1
b	010
c	000
d	011
e	0011
f	0010

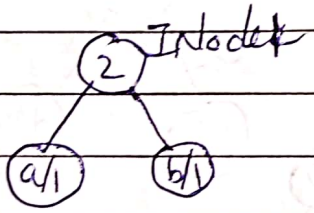
Ans 3: Given, $S = \{a, b, c, d, e, f\}$

$p = \{1, 1, 2, 3, 5, 8\}$

Let us create a minheap:-

a	1
b	1
c	2
d	3
e	5
f	8

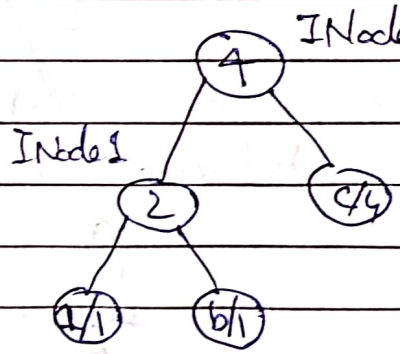
Hence, Now, pop two elements from the top, and make another node that has sum of nodes frequencies as root node.



Minheap:-

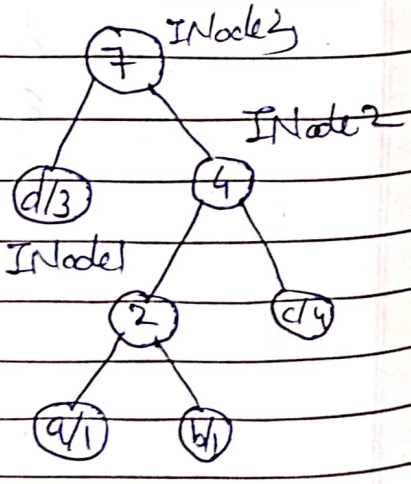
c	2
INode1	2
d	3
e	5
f	8

$\Rightarrow$



Minheap:-

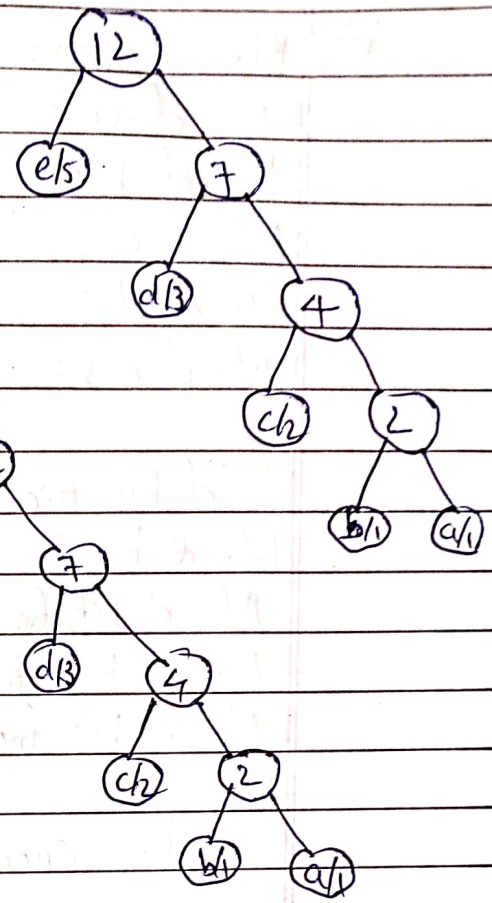
INode2	4
e	5
f	8



Minheap:- ~~I~~ e 5

INode3 7

f 8



Minheap:- INode4 12

f 8

Given, a 1/2

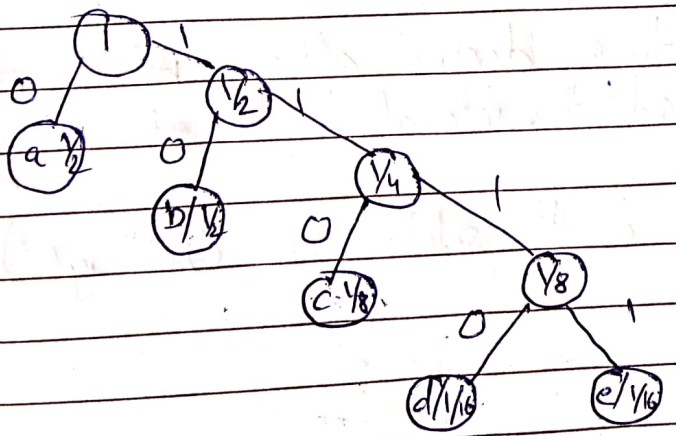
b 1/4

c 1/8

d 1/16

e 1/16

i) The Huffman tree for the given characters and frequencies is:-



Huffman encoding:-

a	0
b	10
c	110
d	1110
e	1111.

b) No. of characters = 1,000,000
No. of a's = 500,000
No. of b's = 250,000
No. of c's = 125000
No. of d's = 62500
No. of e's = 62500

No. of bits for a = 5×10^5 = ~~4×10^6~~ 5×10^5
No. of bits for b = $2 \times 2.5 \times 10^5$ = ~~4×10^6~~ 5×10^5
No. of bits for c = $3 \times 1.25 \times 10^5$ = ~~4×10^6~~ ~~3×10^6~~ 3.75×10^5
No. of bits for d = $4 \times 6.25 \times 10^4$ = ~~4×10^6~~ ~~2×10^6~~ 2.5×10^5
No. of bits for e = $4 \times 6.25 \times 10^4$ = ~~2×10^6~~ 2.5×10^5

Total encoding bit length = ~~15×10^6 = 15 million bits.~~
$= 18.75 \times 10^5$
$= 1.875$ million bits.

Ans 5: The solution needs to be constructed using a greedy approach.
- Sort the intervals given ~~i~~ t in the order of increasing finish times.
- Iterate from 1 to n while maintaining ~~occupied by~~ a list of acceptable tasks.
- If a current task doesnot affect ~~wi~~ any task in the list, then add this interval to the list.

The complexity of this solution is $O(N \log N)$.