* Problem;

⇒ i) N meeting in one room

N meeting ⇒

start → {1, 3, 0, 5, 8, 5}

end → {2, 4, 6, 7, 9, 9}

i) Sorit all meeting with end time of meeting.
Pair of {1, 2}, {3, 4}, {0, 6}, ---, {5, 9}.

when end of first (select) meeting is less than of start of first meeting.

{1, 2} ✓, {3, 4} ✓, {0, 6} ✗, {5, 7} ✓, {8, 9} ✓, {5, 9} ✗

2 < 3 ✓ 4 < 0 ✗ 4 < 5 ✓ 7 < 8 ✓ 9 < 5 ✗

Any -> 4

* ⇒ Shop in candy store

→ find minimum money to buy candy
find maximum money to buy candy

N = 4 , K = 2 ← 1 candy buy and take 2 candy free

C[] = {3, 2, 1, 4}

mini amount:
1 I = [1, 3, 4] (free)
2 I = [1, 2, 3, 4] (Buy)

maxi amount:
4 I = {4, 1, 2} (free)
3 I = {4, 1, 2, 3} (Buy)

* Check if it is possible to survive on island

i/p ⇒ N → Maximum food you can buy per day

S → day to survive

M → require food to survive per day.

Note :- day start ~~from~~ to Monday. and shop is closed Sunday.

o/t ⇒ Find the minimum no of days on which you need to buy food from the shop (कम से कम) so that you can survive the next S days.

⇒

N = 16 , M = 2 , S = 10

1st day ⇒ buy food [16] unit → survive 8 days

9th day ⇒ buy food [16] unit → survive 8 day → [16] days

You need buy twise time food ⇒ [ans - 2]

⇒ N = 20 , M = 30 , S = 10

N < M → return [-1]

Not survive this Condition

→ total food require = S * M

= day of survive * each day require food

Ex ⇒ N = 16 , M = 2 , S = 10

S * M = 20 → S * M % n

20 % 16 = 4

20 / 16 = 1

if ((S * M % n) == 0) then return (S * M % n)

else return ((S * M / n) + 1).

↑ other day food

Ex ⇒ N = 1 , M = 1 , S = 10

S * M = 10

ans = 10 but we can't buy sunday it return -1

↑ One condition get S = 10 ← 1 day sunday

→ buying day = 9

when buying day <= ans

then return -1

* ~~S = i iii~~

Reverse words in a given string

In or S = I like very much
Much very like I

I like very much ← Start

reverse → hcum yrev ekil i
much very like i.

* ## Chocolate Distribution

→ we have M no of Student and given A[] array which have different no of chocolate in Packets.

→ we will be find out maximum no of chocolate and minimum no of chocolate it ~~difference~~ difference.

M=3

A = { 5, 4, 2, 1, 6, 7 }

= { 1, 2, 4, 5, 6, 7 }

3 2

3 2

ans = 2

{ find minimum and maximum then click to sort vector }

* Minimum cost of ropes

Given n ropes

→ merge two ropes and it's cost is sum of length thise ropes.

→ {4, 3, 2, 6}

cost = 5

→ {4, 5, 6}

cost = 9

{6, 9}

cost = 15

29

→ which data structure is given minimum in O(1) complexity.

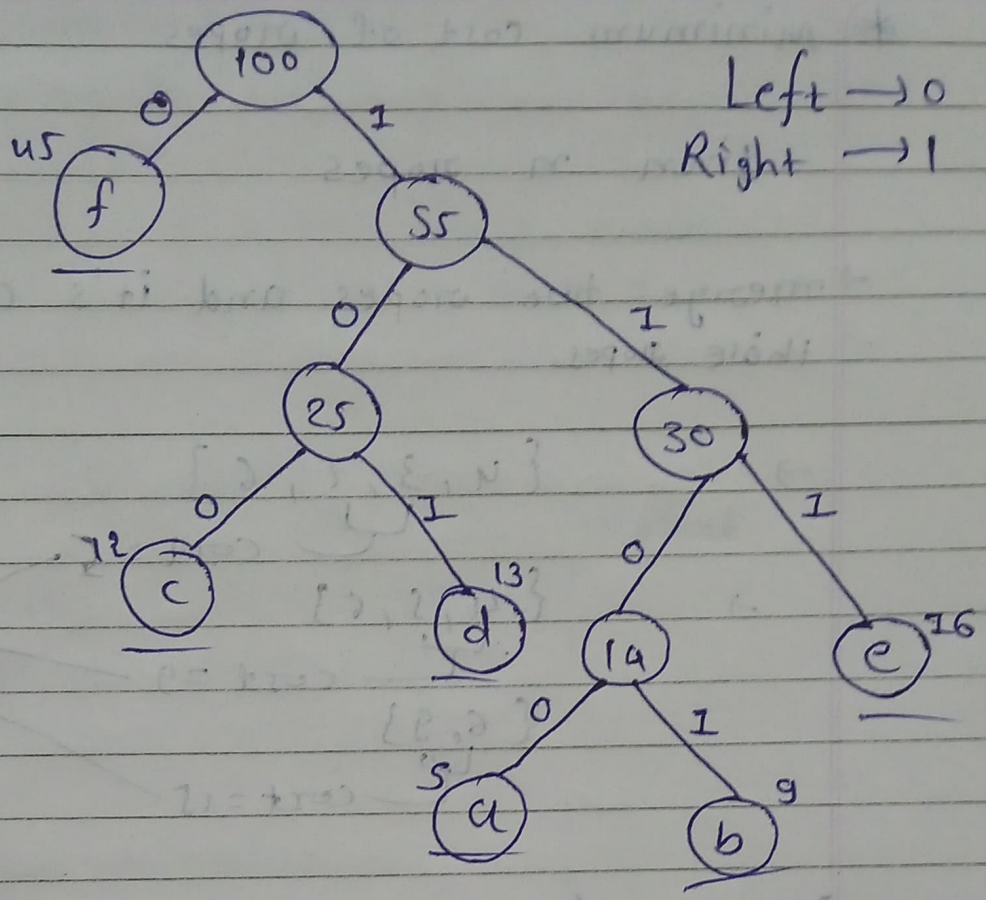
→ min Heap.

* Huffman Encoding

i/p → s = "abcdef"

freq = [5, 9, 12, 13, 16, 45]

→ in huffman encoding, all leaf node is char



Left → 0
Right → 1

f → 0

c → 100

d → 101

a → 1100

b → 1101

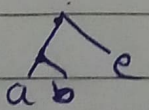
e → 111

a b c d e f

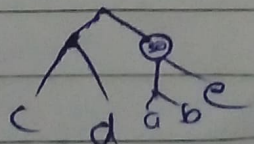
⇒ [5, 9, 12, 13, 16, 45] : (a,5), (b,9)

→ [12, 13, 14, 16, 45] (c,12), (d,13)

→ [14, 16, 25, 45]



→ [25, 30, 45]



→ [45, 55]

* Fractional Knapsack:

Given N items, W require weight

wt → [- - -]

val → [- - -]

→ N = 3 , W = 50

val → [60 , 100 , 120]

wt → [10 , 20 , 30]

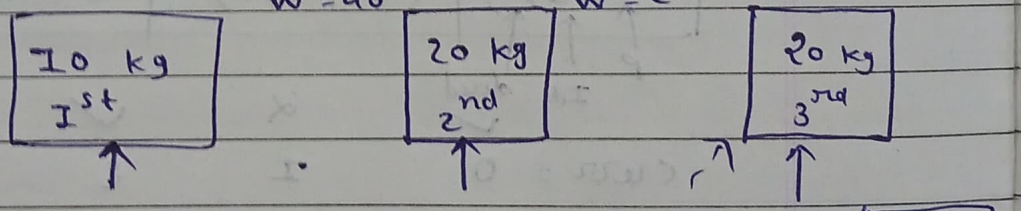
find maximum value

1st	2nd	3rd	
↓	↓	↓	
6 ($\frac{60}{10}$)	5 ($\frac{100}{20}$)	4 ($\frac{120}{30}$)	← value per unit

W = 50

W = 40

W = 20



Fractional part

* Job Sequencing Problem

- O/P → Max profit

Job id	deadline	Profit
1	4	20
2	1	10
3	1	40
4	1	30

curr = 0

0	1	2	3
(40, 3, 1)	(30, 4, 1)	(20, 1, 4)	(10, 2, 1)

p, Id, dead ✓

	✗	✓	✗
curr = 0	1	2	3
Profit = 40	30	20	10
Id = 3	4	1	2
deadline = 1	1	4	1

= 60

Id = 3

1	2	3	4	5	6
~~-1~~ 3	-1	-1	-1 1	-1	-1