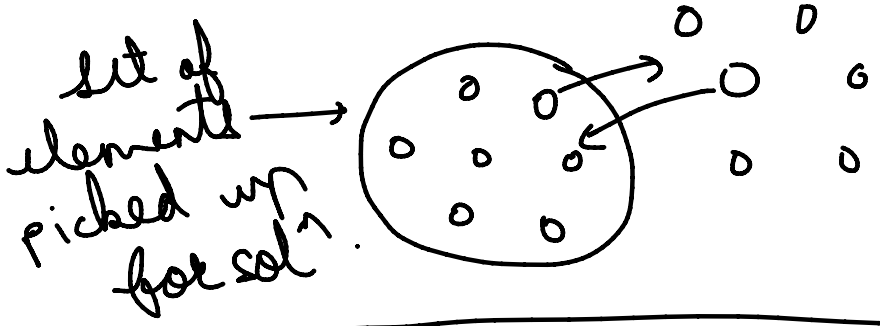
10. Greedy Algorithms

13 July 2024 14:07

⇒ Solving using intuition.

⇒ Choose the best choice available at present.

⇒ Cross verification :-

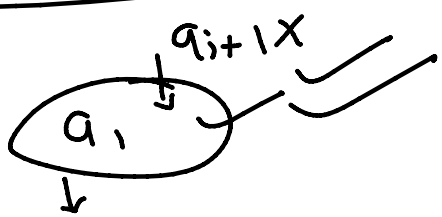


Q. Maximize toys

```
sort(arr);
ans = 0
i = 0 to N-1 :
  if(arr[i] <= K)
  {
    K = K - arr[i];
    ans++;
  }
```

ans => $a_1, a_2 \dots a_i$

$a_1, a_2, a_3 \dots a_i \dots a_N$



Rem → $a_2, a_3 \dots a_i, a_{i+2} \dots a_n$

$K - a[i]$

≥

Rem → $a_2, a_3 \dots a_i, a_{i+2} \dots a_n$

$K - a(i+1)$

Q. Largest number :-

N = 2 S = 16

$N = 2 \quad S = 16$

97 →
88 →
79 →

→ no. of digits (a > b)

10000
9999

10^3 100 10^1 1

→ 98765 4

→ 98900 0

(N) $N = 4, S = 26, 17, 8, 0$

9 9 8 0 ✓

9 9 8 0
9 7 9 9

ans → string

0 → 48
1 → 49
2 → 50
3
⋮
8 → 56
9 → 57

998
99 + '8'
+ char(8 + 48)

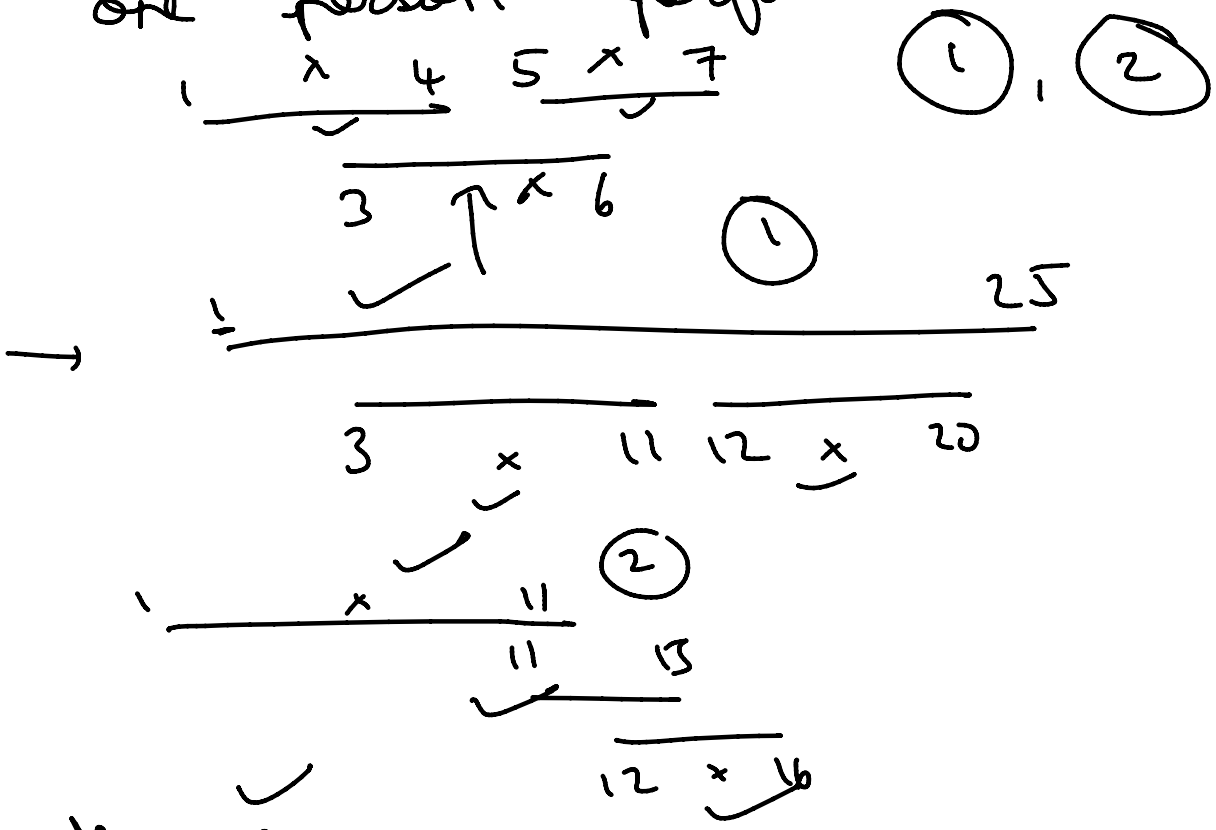
$N = 3$, sum = 30 (N×9)

Q1. Activity Selection Problem:-

N → start[] →
end[] →

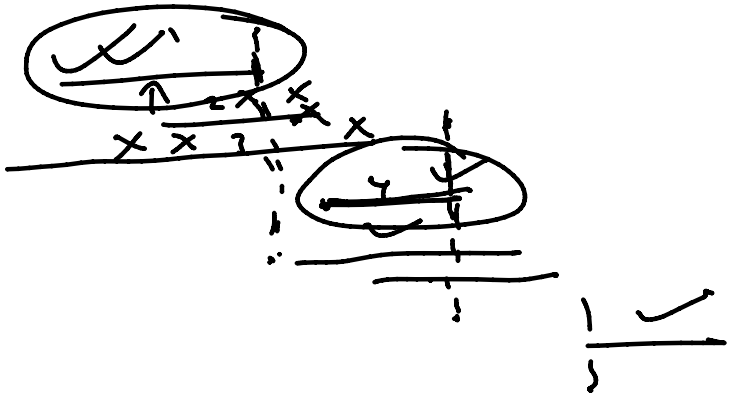
person → performs activities.

one person → performs activities.

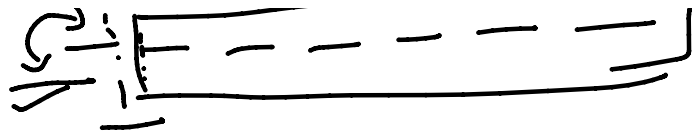


ends → 10:30

ends → 11:00 → sort → write end time



we have not added the min end time activity



min end time activity
✗

better answer

```
if( end[i] < end[j])
{
   b_end[K] = end[i];
   b_start[K] = start[i];
```

3 3 3

3

Q/. a1, a2 → N elements

$\sum a1[i] \times a2[i]$ → minimize

shuffle elements

a1 → asc →
a2 → desc →

$\sum a1[i]\,a2[j]$

1, 4 → 1, 4 → 5+8 = 13
2, 5 → 5, 2 → 2+20 = 22

→ a1, a2 — ai, — aj — an ai < aj
b1, b2 — bi, — bj — bn bi > bj

$$\frac{\sum a[i]b[i] + a[j]b[j]}{\sum a[i]b[j] + a[j]b[i]} \leq$$

$+ \quad \underline{\qquad} \quad \ngeq + a[i]\,b[j] + \dots$

$$a[i]b[j] + a[j]b[i] - a[i]b[i] - a[j]b[j] \geq 0$$

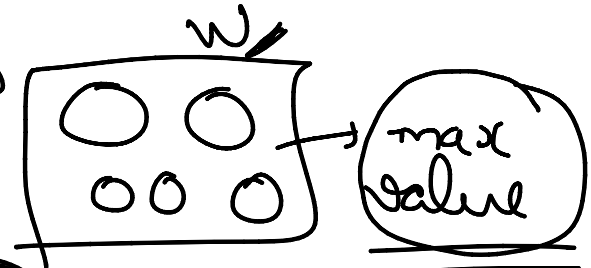
$$a[i]\left(b[j] - b[i]\right) + a[j]\left(b[i] - b[j]\right)$$

$$-a[i]\left(b[i] - b[j]\right) + a[j]\left(b[i] - b[j]\right)$$

$$\underbrace{\left(b[i] - b[j]\right)}_{\geq 0}\underbrace{\left(a[j] - a[i]\right)}_{\geq 0} \geq 0$$

Q/ n items → W → max value

weight ← items → value

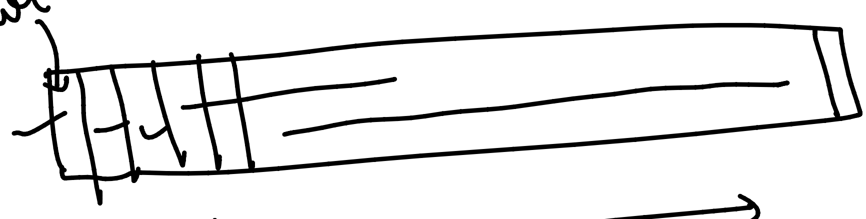


50   20 → 10   w = 25

W = 50   val = [100, 80, 80]   160

(100) $\frac{val}{wt}$ max   wt = [50, 25, 25]

         2   3.2   3.2



sort → val / wt

i → wt = 30, val = 50

w = 15,   (/50) × w

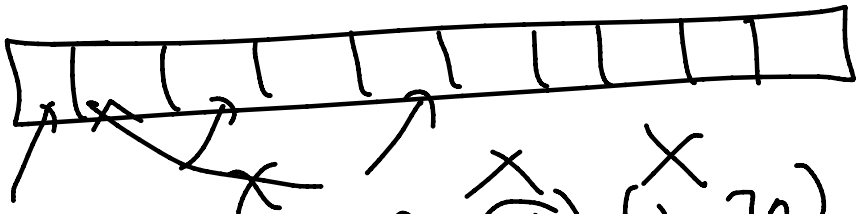
$W = 15$.

$\left(\frac{50}{30}\right) \times W \rightarrow \frac{50}{30} \times 15 = 25$

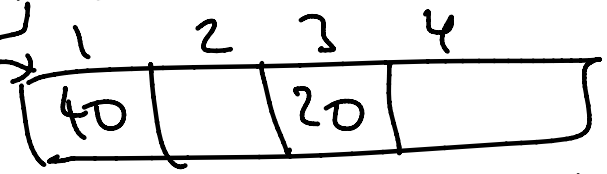
~~$\left(\frac{val}{wt}\right) \times W$~~

Q| Job Sequencing — n jobs.

max profit

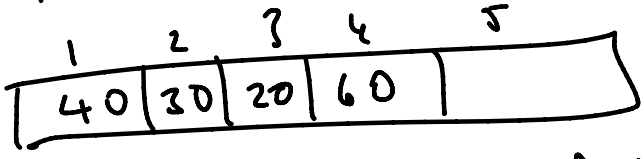


(4, 20) (1, 10) (1, 40) (1, 30)



→ 60, 2

(3, 20), (3, 30), (3, 20), (4, 50), (4, 60)

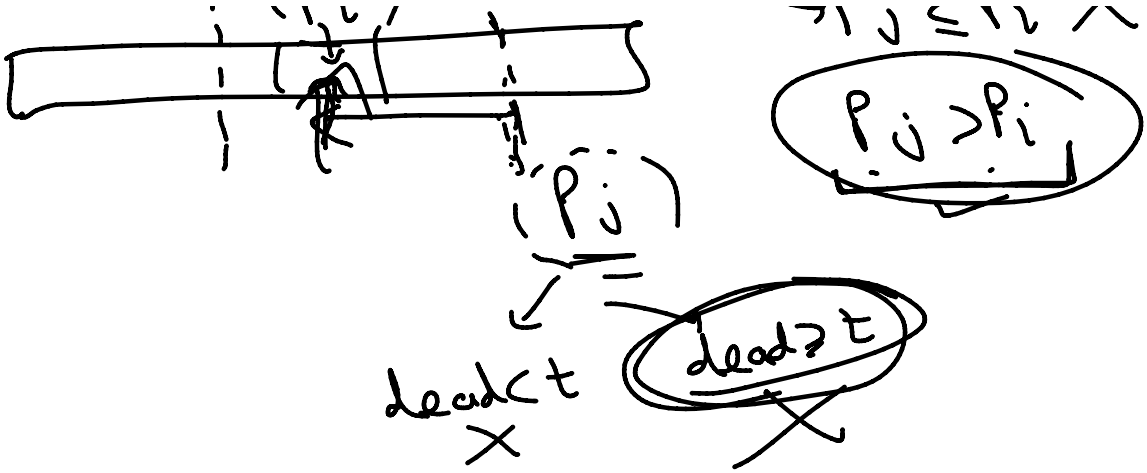


→ sort → Jobs → desc → Profit

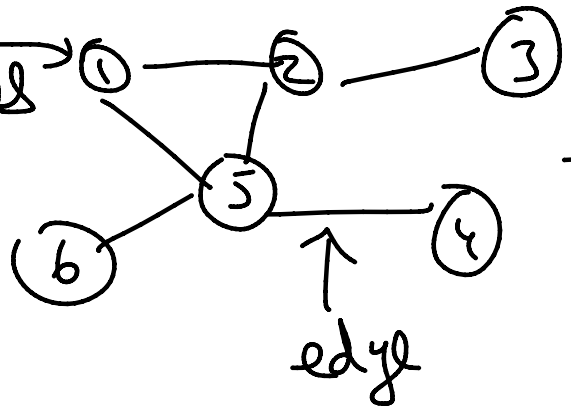
i → deadi

(5) (10)

→ $P_j \leq P_i$ ✗



⇒ Graphs :- nodes / vertices



⇒ 5,6 are adjacent

edge

→ graph set of vertices & edges

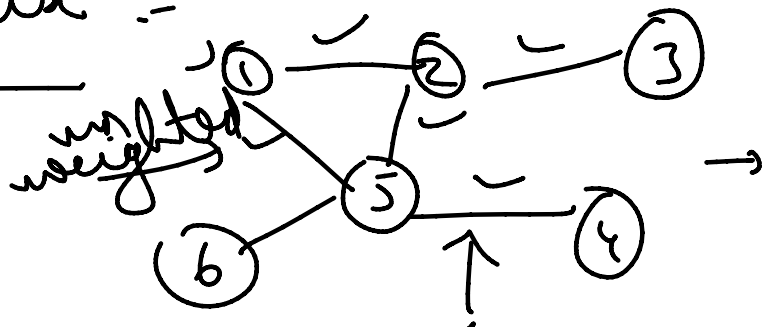
⇒ Storing of graph :-

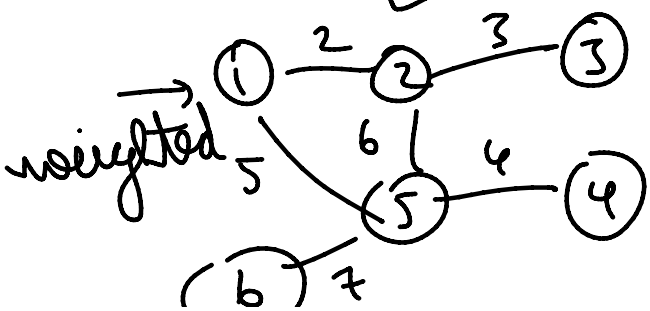
① Adjacency matrix :-

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	0	0	0
4	0	0	0	0	1	0
5	1	1	0	1	0	1
6	0	0	0	0	1	0

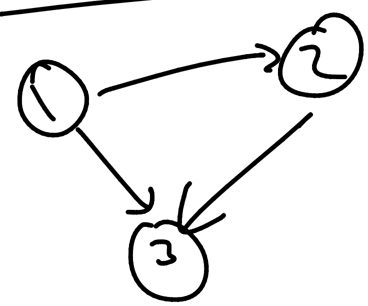
$S = O(N^2)$

Directed graph





Directed graph



	1	2	3
1	0	1	1
2	0	0	1
3	0	0	0

Adv :-

① Removing / adding an edge is $O(1)$

② Detecting an edge is $O(1)$.

→ **Disadv :-**

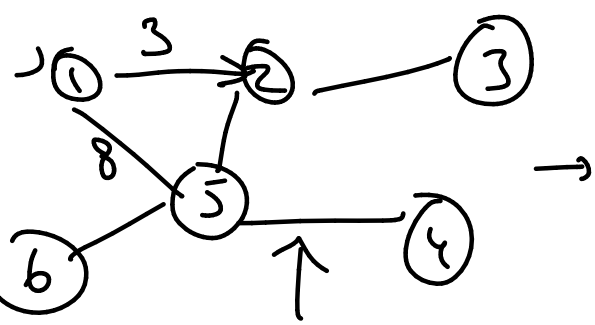
① Wastage of space in case of sparse matrix.

② Traversal of graph is costly.

⇒ ② **Adjacency list :-** ↳ {⟨2,3⟩, ⟨5,8⟩}

1 → 2, 5
2 → 1, 3, 5
3 → 2
4 → 5
5 → 1, 2, 4, 6
6 → 5

$S = O(V + E)$



Adv :- ① The space wastage

② [illegible] traversal

Adv :- ① No space wastage
② Less time in graph traversal

Disadv :-
① Detect an edge takes $O(E)$ time
② Removing an edge takes $O(E)$ time.