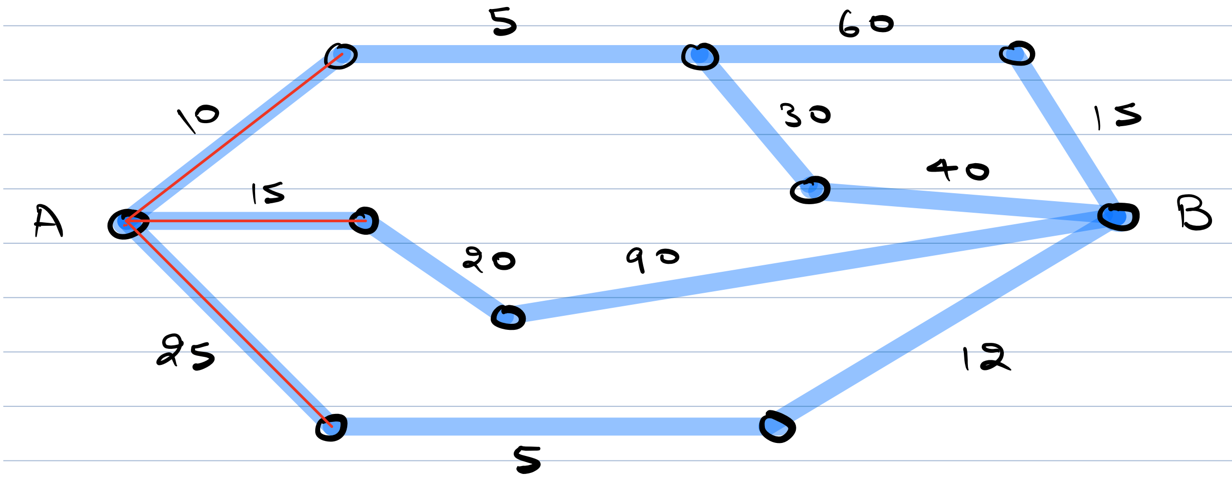
1) Greedy
2) When to apply greedy.
3) Problems.

Recruiter Google	Jan	Feb	Mar
	[✓ (40L)]	(40L)	(41L)
	(50L)	(56L)	(36L)
	(55L)	(67L)	(38L)
		[✓ (35L)]	[✓ (35L)]

Goal : Reduce the overall annual hiring cost.

1) Best choice locally leads to best global solution

2) Current local choice will not affect any future decisions.



* If greedy fails, explore all possibilites
(Recursion)

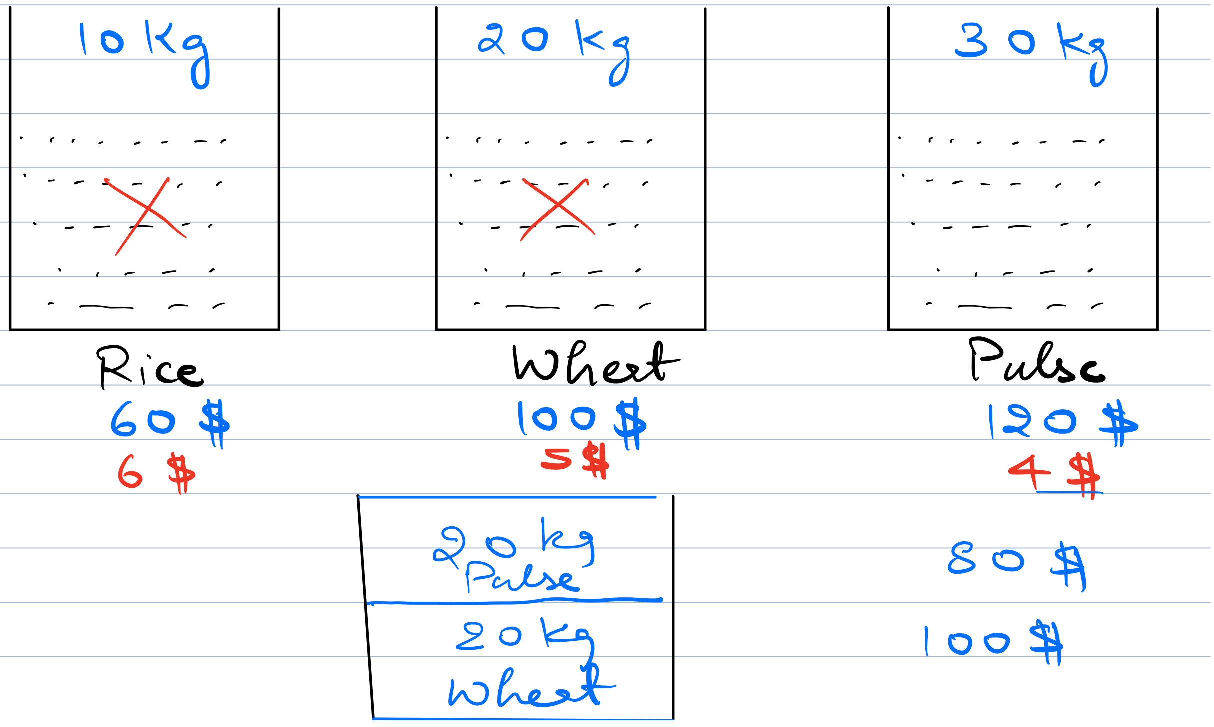
Q. Fractional Knapsack

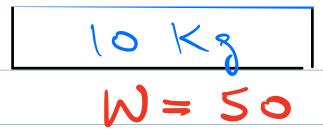
Given N items → Weight
→ Cost (value)

Given a bag with weight capacity W

Pick some items s.t. the value stored in the bags is maximised / minimised.

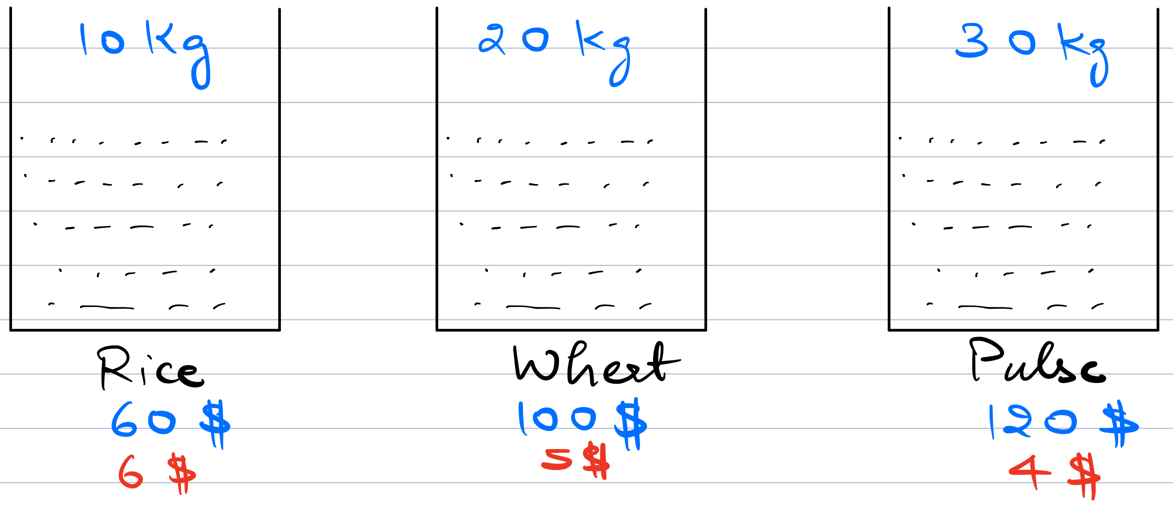
Super Market

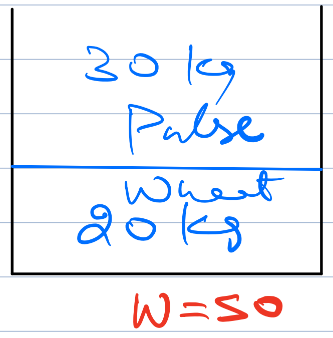




60 $
240 $

Const. : Either select all or selett nothing





220 $

0/1 Knapsack → DP

Steps :

1) Sort all the items on the basis of per unit cost

2) Iterate, pick the best & keep updating the seg capacity.

$$T.C. = O(N \log N + N)$$
$$= O(N \log N)$$

Q// Anirudddha (CEO) (famous)

BMW approaches Anirudddha Special sale.

N cars.

A ⇒ A[i] = Time till which i^{th} car is free of cost (sale is active)

B ⇒ B[i] = Beauty of i^{th} car.

⇒ Maximize the overall beauty value of the cars purchased.

⇒ It takes 1 unit of time to purchase 1 car.

	✓		✓		✓
	0	1	2	3	4
A ⇒	[3	1	3	2	3]
B ⇒	[6	5	2	1	9]
	×	×	×	×	×

T= 0	9	5
T = 1	6	6
T = 2	3	9
T = 3	18	20 Ans

1) Sort the cars on the basis of the sale end time

A :	1	5	5	5	3	3	3
B :	5	4	3	8	20	7	10

	0	1	2	3	4	5	6
A :	1	3	3	3	5	5	5
B :	5	20	7	10	4	3	8 ↑

T = 0	~~0 (5)~~	3 (10)
T = 1	1 (20)	
T = 2	2 (7)	
T = 3	4 (4)	
T = 4	~~5 (3)~~	6 (8)
T = 5		

min Beauty

⇓

Min Heap

Code

1) Sort the cars (H.W.)

```
int t = 0;
min Heap m;
```

```
total Beauty = 0;
for (i = 0; i < N; i++) {
        if (t < A[i]) {
                m.insert(B[i]);
                totalBeauty += B[i];
                t++;
        }
        else {
                if (B[i] > m.peek()) {
                        totalBeauty -=
                                m.deleteMin();
                        m.insert(B[i]);
                        totalBeauty += B[i];
                }
        }
}
```

$T.C. = O(N \log N + N \log N)$

$= O(N \log N)$

$S.C. = \underline{O(N)}$