Greedy Method →

The greedy method is used for solving optimization Problem.

→ Optimization Problem is a problem that demands or requires maximum profit and minimum cost.

e.g. Problem P : city A → city B.

Problem P is to travel from city A to city B

For this problem there can be more than one solution.

S_1 by walk
S_2 by car
S_3 by byke
S_4 by bus
S_5 by train
S_6 by flight

} Solution space.

But suppose we have a constraint that the travel time is to be 12 hrs.

$$P : A \xrightarrow{12 \text{ hrs}} B$$

This contraint is solved by S_5 and S_6. This type of solutions are called feasible solutions (solutions that satify the constraints)

S_5
S_6] Feasible Solution.

Now suppose I want to cover this journey in minimum cost. So this becomes a minimization problem

Out of S_5 and S_6, S_5 takes less cost to go to city B

So this is called optimal solution.

One optimal solution exist for any problem.

If a problem requires either minimum result or maximum result, is called optimization Problem. ②

Strategies used for solving optimization Problem →

1. Greedy Method
2. Dynamic Programming
3. Branch and Bound.

Application of Greedy Method —

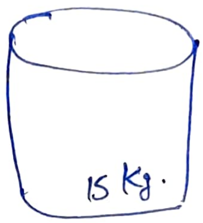
1. Knapsack Problem
2. Job sequencing with deadline
3. min. cost spanning tree
   - Krushkal
   - Prims
4. optimal Merge Pattern
5. Huffman coding
6. Single source shortest Path.
   - Dijkstra
   - Bellman ford.

Knapsack Problem → (Fractional 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
$\frac{P}{W}$	5	1.3	3	1	6	4.5	3

$n = 7$
$m = 15$

Some objects are given, every object is having some profit associated with it and every object is having some weight.



We have to fill this bag with these objects and we will carry this bag to different place and we will get some profit. The problem is container loading problem.

no of objects $= n$
capacity of knapsack $= m$

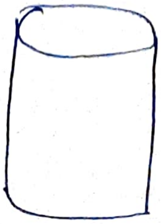
So problem is $\sum_{i=1}^{n} W_i > m$

Constraint is $\sum_{i=1}^{n} x_i w_i \leq m$.

objective is Maximize Profit
$\sum_{i=1}^{n} x_i P_i$ is Max

x (0–1
x_1 x_2 x_3 x_4 ...)

$0 \leq x \leq 1$ (i.e this knapsack problem is for objects which can be taken in fractions)



5 $15-1=14$
$14-2=12$
$12-4=8$
$8-5=3$
$3-1=2$
$2-2=0$

1	2/3	1	0	1	1	1
x_1	x_2	x_3	x_4	x_5	x_6	x_7

4

$$\sum_{i=1}^{n} x_i w_i = 1\times2 + \frac{2}{3}\times3 + 1\times5 + 0\times7 + 1\times1 + 1\times4 + 1\times1$$

$$2+2+5+0+1+4+1 = 15$$

$$\sum x_i P_i = 1\times10 + \frac{2}{3}\times5 + 1\times15 + 1\times6 + 1\times18 + 1\times3$$

$$10 + 2\times1.3 + 15 + 6 + 18 + 3 = \underline{54.6}$$

1. Given a list of n objects
2. Capacity of knapsack is m
3. Each object I_i has a weight w_i and a profit P_i

   $w_i > 0$ $P_i > 0$

4. In greedy knapsack problem, items can be broken into smallest pieces.
5. If a fraction x_i ~~belongs to~~ $\{x_i \in 0 \ldots 1\}$ of an object I_i is placed into a knapsack than profit $P_i x_i$ is earned.
6. objective of algo is to maximize the profit.

	1	2	3	4	5
Profit	30	20	100	90	160
weight	5	10	20	30	40
P/w	6 (1)	2 (5)	5 (2)	3 (4)	4 (3)

$m = 60$, 30

1 - 6 - 5
2 - 5 - 20 } 25
3 - -

$\frac{30}{5} = 5$

1	2	3	4	5
1	0	1	0	.875

$\frac{40\times5}{100}$

```
                                  w[1...n]
Knapsack(P[1...n], x[1...n], m)

For i=1 to n
do P(i)/w(i)
Sort decreasing order
i=1
while (weight < m)
{
    if (weight+w[i] ≤ m)
        x[i]=1
        weight=weight+w[i]
    else
        x[i]= (m- weight) % w[i]
        weight = m.
Profit = Profit P[i] * x[i]
i++;
}
```

weight=0
Profit =0
i=1
(0<60)
if(0+5) ≤ 60
weight = 0+5=5
p=0+30×1=30
i=2

```
For (i=1 to n)                                   O(n)
{ compute Pi/wi;
}
 Sort objects in non increasing order of P/w _ O(n log n)
   For i=1 to n
       if m>0 && wi ≤ m
           m=m-wi                                 O(n)
           p=p+Pi
       else break;
   if m>0
       P= P+ Pi(m/wi) — O1
```