**GREEDY METHOD**

Greedy algorithms are used where we are not concerned with optimal solutions. However, in certain cases greedy algorithm can solve some of the well known problem that effectively which give the optimal result but they after provide fast non-optimal strategies. Which are helpful in obtaining good approximations?

**Steps for greedy method:-**

1. **Feasible:** here, we check whether it satisfied all possible problem constraint or not, so as to obtain at least one solution to our problem.
2. **Local optimal choice:** - optimum which is selected from current available feasible choices.
3. **Unalterable:** - once the choice is made at any sub spent step that choice is not altered.

Knapsack problem:-   
 the knapsack problem into pack the knapsack with maximum value in such a manner that total weight of the items should not be greater than capacity of knapsack.

We can have three criterions for this problem.

1. In this criterion items are arranged by their values. Here the item with maximum value is selected first and process continues till minimum value.
2. In this criterion, “items are arranged by their weights. Here the item with lightest weight is selected first & process continues till the maximum weight.
3. In this criterion, items are arranged by certain ratio, Pi --🡪ratio of value are weight. Here selection proceeds from maximum ratio to minimum ratio.

After selecting criteria, we check whether the weight Wi , of item Ii from top of list in less than knapsack capacity is more than or equal.

**ALGORITHM:**

**Algorithm knapsack(I, W, V)**

//algorithm determines the maximum value that a knapsack can have after sorting items with total weight less than or equal to capacity of knapsack.

1. Set S🡨{ phy }
2. Set V🡨O
3. For i🡨I to n
4. If (WI<=W) then
5. Set S🡨Ii //adding items
6. Set V🡨V+VI //adding value of ith term
7. Set W🡨w - wi //new maximum weight
8. Return S & V.

Q: consider five items along with their respective weights and values.

I = <I1, I2, I3, I4, I5>

W = < 5, 10. 20, 30, 40>

V = <30, 20, 100, 90, 160>

Knapsack capacity W=60.

SOLUTION: Initially,

|  |  |  |
| --- | --- | --- |
| Items | Wi | Ii |
| I1 | 5 | 30 |
| I2 | 10 | 20 |
| I3 | 20 | 100 |
| I4 | 30 | 90 |
| I5 | 40 | 160 |

1. Take value/weight ratio.

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Wi | Vi | Pi=Vi/Wi |
| I1 | 5 | 30 | 6 |
| I2 | 10 | 20 | 2 |
| I3 | 20 | 100 | 5 |
| I4 | 30 | 90 | 3 |
| I5 | 40 | 160 | 4 |

Arrange items with decreasing order of Pi.

|  |  |  |  |
| --- | --- | --- | --- |
| ITEM | Wi | Vi | Pi=Vi/Wi |
| I1 | 5 | 30 | 6 |
| I2 | 20 | 100 | 5 |
| I3 | 40 | 160 | 4 |
| I4 | 30 | 90 | 3 |
| I5 | 10 | 20 | 2 |

Now we pick value I1, I3 as maximum value=V1+V3=100+30=130

Wi=WI+W3=20+5=25

But the total capacity of knapsack =60

So, we take W-Wi=35 from next item. Then, 35 weight required will be taken from next ITEM1. I5 (having less”P” ratio)

Now Pi=100+30+140=270

WI=20+5+35=60

So maximum value or profit=270