GREEDY METHOD

EXPERIMENT : No.

PAGE No.

DATE

Structure of Greedy Algorithm

- It is simple, intuitive algorithm that is used in optimization problems.

- It takes all of the data in a particular problem, and then set a rule for which elements to add to the solution at each step of the algorithm.

~~Properties of problems~~

If properties below are True, then greedy algorithm can be used to solve the problem.

Greedy choice property:

- A global optimal solution can be reached by choosing the optimal choice at each step.

Optimal Substructure:

- A problem has an optimal substructure if an optimal solution to the entire problem contains the optimal solutions to the subproblem.

In other words,

" It works on problems for which it is true that, at every step, there is a choice i.e optimal for the problem upto that step after last step algorithm produces globally optimal solution".

Teacher's Sign.:

GENERAL METHOD :

- Given 'n' inputs choose a subset that satisfies some constraints.
- A subset that satisfies the constraint is called feasible solution.
- A feasible solution that maximizes or minimizes a given (objective) function is said to be optimal.
- Often, it is easy to find feasible solution but difficult to find the optimal solution.

" The greedy method suggests that one can devise an algorithm that works in stage. At each stage, a decision is made whether a particular input is in the optimal solution. This is called Subset paradigm."

GENERAL ALGORITHM STRUCTURE :

```
Algorithm Greedy (A, n)    A -> set, n -> integer
{
1.    MakeEmpty (solution)
2.    for ( i = 1 to n ) {
3.        x = Select (A)           x -> one input
4.        if Feasible (solution, x) then
5.            solution = Union(solution, x)
      } // End of for.
6.    return solution.
}
```

EXPERIMENT : No. DATE

Select (Objective) - selects an input from A whose value is assign to x.

Feasible (Constraint) - Boolean valued function that determines if 'x' can be included into the solution vector.

Union (Building Solution) - combines x with the solution and update the objective function.

EXPERIMENT : No. FRACTIONAL KNAPSACK

PAGE No. | DATE

(GREEDY METHOD)

I] Problem Definition :

- Given n objects and a knapsack with a capacity (weight) M.
- Each object 'i' is associated with weight w_i and profit p_i.
- For each object i, suppose a fraction x_i, $\boxed{0 \leq x_i \leq 1}$ (i.e 1 is the maximum amount) can be placed in the knapsack, then profit earned is, $= p_i x_i$
- Objective: To maximize profit subject to capacity constraint
- i.e.

Maximize $$\sum_{i=1}^{n} p_i x_i \quad \text{——— (1)}$$

subject to $$\sum_{i=1}^{n} w_i x_i \leq M \quad \text{——— (2)}$$

where
$$\left.\begin{array}{l} 0 \leq x_i \leq 1, \\ p_i > 0 \\ w_i > 0 \end{array}\right\} \quad \text{——— (3)}$$

- A feasible solution is any subset $\{x_1, x_2 \dots x_n\}$ satisfying equation (2) and (3)
- An optimal solution is a feasible soln that maximizes (1)

Teacher's Sign.:

Scanned by CamScanner

II] Application :

Knapsack problems appear in real world decision making processes in a wide variety of fields, such as,

(i) Finding the least wasteful way to cut raw-materials,

(ii) Selection of investments and portfolios.

(iii) Resource Allocation

(iv) Container ~~Bo~~ Loading etc.

III] Algorithm :

Input : M : Knapsack Capacity.

n : Number of objects.

p[1..n] : } → contains the profits and weights
w[1..n] : } respectively of the n object ordered such that $\boxed{p[i]/w[i] \geq p[i+1]/w[i+1]}$

Select

Output : x[1..n] → Solution vector

```
Algorithm   GreedyKnapsack (m,n)
  {
1.       for i = 1 to n
2.             x[i] = 0.0          // initializing solution
                                       Vector
            Rem
3.       capacity = m
4.     { for   i = 1   to   n          // select
5.             if (w[i] > Rem capacity)   // Feasible
6.                   break
7.             x[i] = 1                 // union
```

EXPERIMENT : NO. DATE

8. Rem Capacity = Rem capacity − w[i]

} // End of for

9. if (i ≤ n)

10. x[i] = Rem capacity / w[i]

Remaining Capacity.

}

IV Analysis :

For greedy knapsack algorithm, line 1 and line 4 will ~~take~~ executes (n+1) time.

After analyzing with table method, we will get a polynomial of degree 1.

∴ Running time complexity i.e. $T(n) = \Theta(n)$ or $O(n)$ or $\Omega(n)$.

But if knapsack is not sorted, then sorting algorithm is also required then

Running time complexity i.e

$$T(n) = \underbrace{\Theta(n\log n)}_{\text{sorting}} + \underbrace{\Theta(n)}_{\text{greedy knapsack}}$$

∵ first term is larger than second term

$$\therefore \boxed{T(n) = \Theta(n\log n)}$$

V] **Example:**

A thief enters a house for robbing it. He can carry a maximal weight of 60kg into his bag. There are 5 items in the house with the following weights and values. What items should thief take if he can even take the fraction of any item with him?

Item	Weight	Value (profit)
I1	5	30
I2	10	40
I3	15	45
I4	22	77
I5	25	90

Solution:

Step 1: Compute profit / weight ratio

Items	Weight	Profit	Ratio
I1	5	30	6
I2	10	40	4
I3	15	45	3
I4	22	77	3.5
I5	25	90	3.6

Step 2: Sort all the items in decreasing order of Ratio.

Items	Ratio
I1	6
I2	4
I5	3.6
I4	3.5
I3	3

EXPERIMENT : No. PAGE NO. DATE

Step 3 : Greedy Knapsack.

Iteration	Knapsack wt	Items in Knapsack	~~Total~~ profit
0	60	ϕ	0
1	55	{I1}	30
2	45	{I1, I2}	70
3	20	{I1, I2, I5}	160

Now, $i = 4 < n$

$\therefore$ Total profit $= 160 + \cancel{20}\ \frac{capacity}{w[i]}\ P[i]$

$= 160 + \frac{\cancel{20}\ 10}{\cancel{22}\ 2}\ \boxed{\cancel{77}}\ 7$

$= 160 + 70 = 230$

$\therefore$ Solution vector $= x = [1, 1, 0, \frac{20}{22}, 1]$

I1 I2 I3 I4 I5

Example [2]

W = 60 (knapsack capacity)

Items	Profit	Weight
A	280	40
B	100	10
C	120	20
D	120	24

Solution :-

step 1 : Compute profit / weight Ratio

Items	Profit	Weight	Ratio
A	280	40	7
B	100	10	10
C	120	20	6
D	120	24	5

step 2 : Sort all the items in decreasing order of Ratio

Items	Ratio
B	10
A	7
C	6
D	5

step 3 : Greedy knapsack

Iteration	knapsack wt	Items in knapsack	Total profit
0	60	ϕ	0
1	50	{B}	100
2	10	{B, A}	380
3			

EXPERIMENT : No.

PAGE No.
DATE

Now, $i = 3 < n$

Rem capacity

profit of C

$\therefore$ Total Profit $= 380 + \frac{10}{20}[120] = 440$

wt of C

$\therefore$ Solution vector $= \begin{array}{cccc} \begin{bmatrix} 1 & 1 & 1/2 & 0 \end{bmatrix} \\ A \quad B \quad C \quad D \end{array}$