Department of Computer Science Engineering B.TECH, V-SEM
SUBJECT:- DAA
TOPIC:- Greedy Method (Unit-2)
By:- Mr. Dinesh Swami

Greedy Method Algorithms

Greedy algorithm :-

Greedy algorithm is an algorithm that solves the problem by taking optimal solution at the local level (without regards for any consequences) with the hope of finding optimal solution at the global level.

Greedy algorithm is used to find the optimal solution but it is not necessary that you will definitely find the optimal solution by following this algorithm.

1. Knapsack Problem

2. Job Sequencing

3. Optimal Merge Patterns

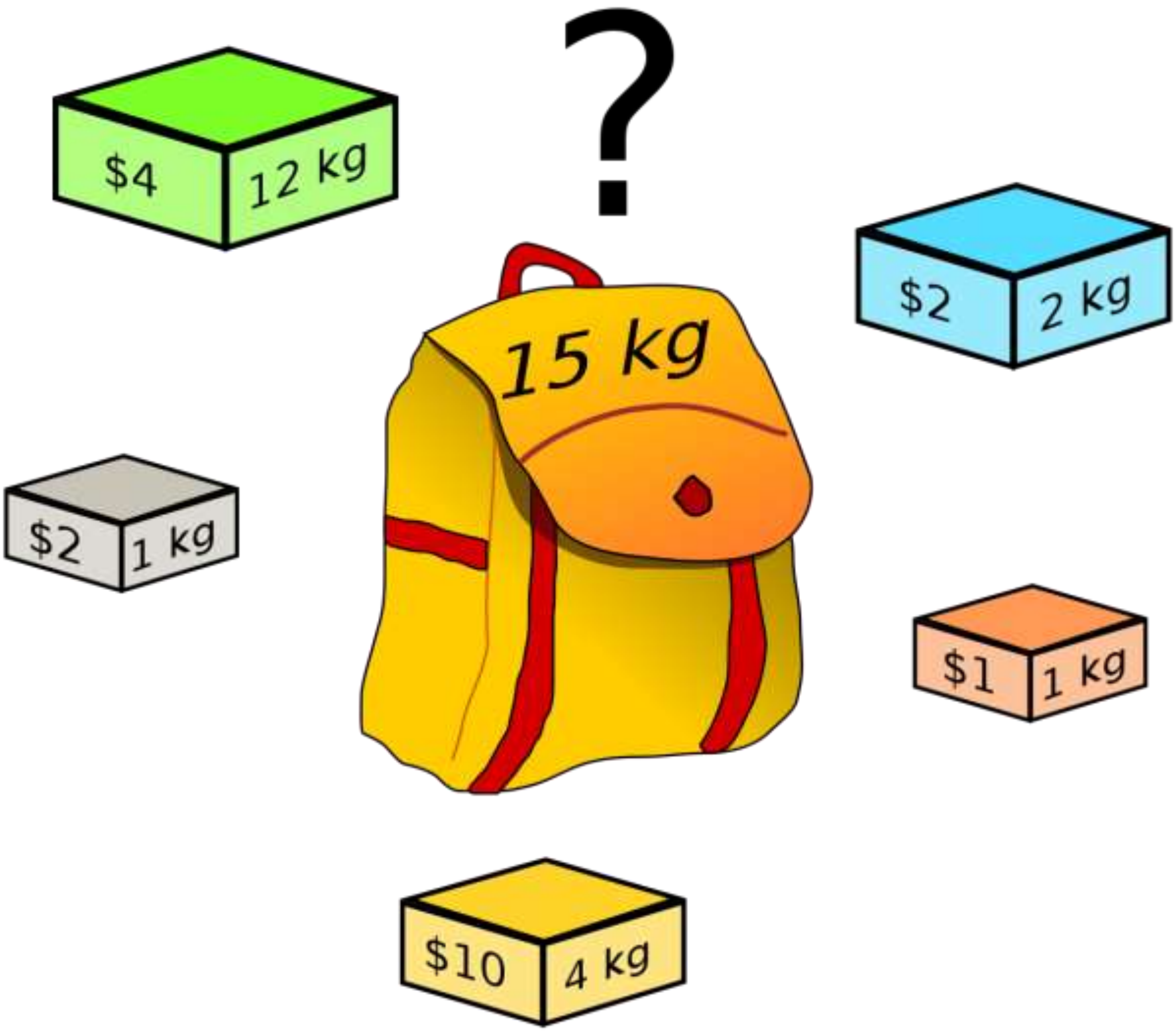
4. Minimal Spanning Tree

1. Fractional Knapsack

Fractions of items can be taken rather than having to make binary (0-1) choices for each item.

Fractional Knapsack Problem can be solvable by greedy strategy whereas 0 - 1 problem is not.

The basic idea of the **greedy approach** is to calculate the ratio value/weight for each item and sort the item on basis of this ratio. Then take the item **with** the highest ratio and add them until we can't add the next item as a whole and at the end add the next item as much as we can.



Pi= Vi / Wi=

G= 4/12=0.33

Sky=2/2=1.0

Black=2/1=2.0

O= 1/1=1.0

Yellow= 10/4= 2.5

15-> 4kg(y)+1kg(black)+2kg(sky)+1kg(O)+ 7kg(g) 10$+2$+2$+1$+(7*0.33=   2.31$)=17.31$

Steps to solve the Fractional Problem:

1. Compute the value per pound Vi/Wi for each item.
2. Obeying a Greedy Strategy, we take as possible of the item with the highest value per pound.
3. If the supply of that element is exhausted and we can still carry more, we take as much as possible of the element with the next value per pound.
4. Sorting, the items by value per pound, the greedy algorithm run in O (n log n) time.

```
Fractional Knapsack (Array v, Array w, int W)
for i= 1 to size (v)
do p [i] = v [i] / w [i]
Sort-Descending (p)
i ← 1
while (W>0)
do amount = min (W, w [i])
solution [i] = amount
W= W-amount
i ← i+1
return solution
```

Example: Consider 5 items along their respective weights and values: -

$I = (I_1, I_2, I_3, I_4, I_5)$

w = (5, 10, 20, 30, 40)

v = (30, 20, 100, 90,160)

The capacity of knapsack W = 60

Now fill the knapsack according to the decreasing value of p_i.

First, we choose the item I_i whose weight is 5.

Then choose item I_3 whose weight is 20. Now,the total weight of knapsack is 20 + 5 = 25

Now the next item is I_5, and its weight is 40, but we want only 35, so we chose the fractional part of it, i.e., $5 \times \frac{5}{5} + 20 \times \frac{20}{20} + 40 \times \frac{35}{40}$

Weight = 5 + 20 + 35 = 60

Maximum Value:-

$30 \times \frac{5}{5} + 100 \times \frac{20}{20} + 160 \times \frac{35}{40}$

= 30 + 100 + 140 = 270 (Minimum Cost)

Solution:

ITEM	w_i	v_i
I_1	5	30
I_2	10	20
I_3	20	100
I_4	30	90
I_5	40	160

Taking value per weight ratio i.e. $p_i = v_i / w_i$

ITEM	w_i	v_i	$P_i = v_i / w_i$
I_1	5	30	6.0
I_2	10	20	2.0
I_3	20	100	5.0
I_4	30	90	3.0
I_5	40	160	4.0

Now, arrange the value of p_i in decreasing order.

ITEM	w_i	v_i	$p_i = v_i / w_i$
I_1	5	30	6.0
I_3	20	100	5.0
I_5	40	160	4.0
I_4	30	90	3.0
I_2	10	20	2.0

i.e., $5 \times \frac{5}{5} + 20 \times \frac{20}{20} + 40 \times \frac{35}{40}$

Weight = 5 + 20 + 35 = 60

Maximum Value:-

$30 \times \frac{5}{5} + 100 \times \frac{20}{20} + 160 \times \frac{35}{40}$

= 30 + 100 + 140 = 270 (Minimum Cost)

Fractional Knapsack Problem (Greedy Algorithm): Example 2

Knapsack weight = 5 lb.

Item	1	2	3	4
Value, $	12	10	20	15
Weight, lb	2	1	3	2

Solution: Compute the Value/Weight for each item

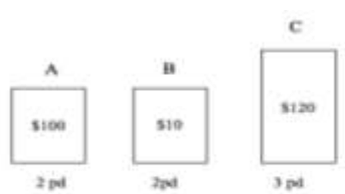
Item	1	2	3	4
Value/Weight	6	10	6.67	7.5

Re-ordering the items according to the decreasing order of Value/Weight (break the tie by picking the item with the lowest Index)

Item	2	4	3	1
Value/Weight	10	7.5	6.67	6
Value, $	10	15	20	12
Weight, lb	1	2	3	2
Weight collected	1	2	2	

Items collected: Item 2 (1 lb, $10); Item 4 (2 lb, $15); Item 3 (2 lb, (2/3)*20 = $13.3);
Total Value = $38.3

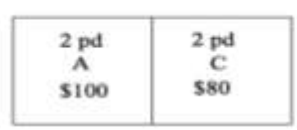
Example :-3



Capacity of knapsack: $K = 4$

Fractional Knapsack Problem:
Can take a fraction of an item.

Solution:



2. Job Sequencing:-

In job sequencing problem the objective is to find the sequence of jobs, which is completed within their deadlines and give maximum profit.

If a set of ***n*** jobs are given which are associated with deadlines and profit is earned and a job is completed by its deadline. These jobs need to be ordered in such a way that there is maximum profit.

Job Sequencing with Deadlines:-

The problem is stated as below.

- There are n jobs to be processed on a machine.
- Each job i has a deadline $d_i \geq 0$ and profit $p_i \geq 0$.
- Pi is earned iff the job is completed by its deadline.
- The job is completed if it is processed on a machine for unit time.
- Only one machine is available for processing jobs.
- Only one job is processed at a time on the machine.

To solve the job sequencing problem via greedy method follow this steps:

1. *Sort all jobs in decreasing order of profit.*
2. *Initialize the result sequence as first job in sorted jobs.*
3. *Do following for remaining n-1 jobs.*
4. *If the current job can fit in the current result sequence without missing the deadline, add current job to the result. Else ignore the current job.*

Example-1 :-

n=5

Jobs :-	**J1**	**J2**	**J3**	**J4**	**J5**
Profits:-	**20**	**15**	**10**	**5**	**1**
Deadlines:-	**2**	**2**	**1**	**3**	**3**

J1-> J2 -> J4

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

J1-→J2--→J3

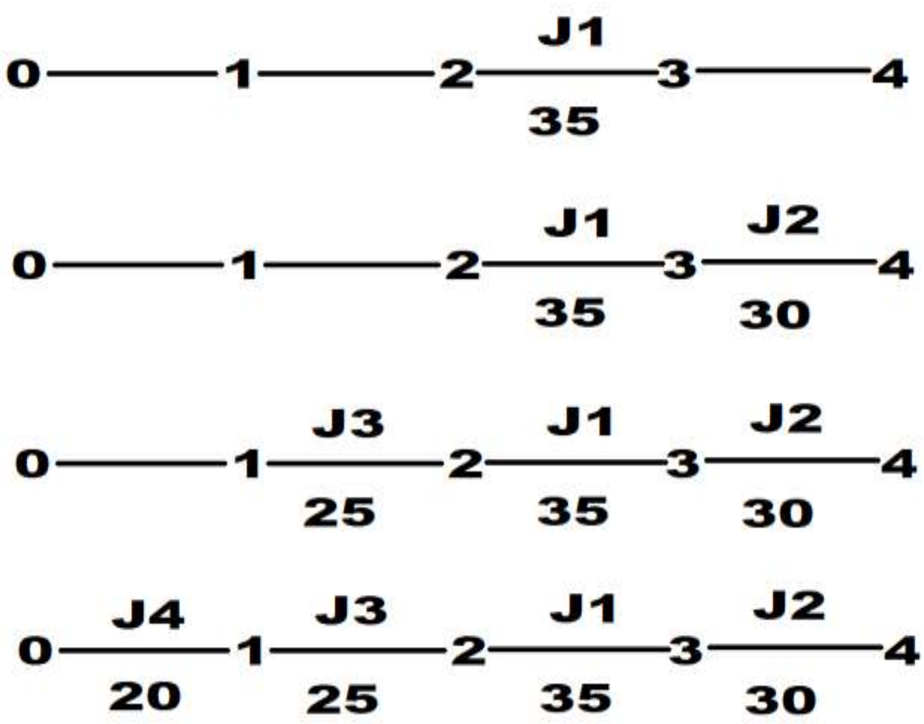
20+15+5=40

Example-2:-

n=7

Jobs	J1	J2	J3	J4	J5	J6	J7
Profits	35	30	25	20	15	12	5
Deadlines	3	4	4	2	3	1	2

SOLUTION:-



THE ORDER OF SEQUENCING:-

Here job 1 has the highest profit and the deadline is 3 so set that job between 2–3 .Job 2 has second highest profit and the deadline is 4 so set that job between 3–4 and do same to all jobs.

So the final sequence is J4-J3-J1-J2 and the total profit is 110.

OR

<u>OTHER ORDER OF SEQUENCING:-</u>

```
  J1     J4       J2      J3            35+20+30+25= 110
0-----1-------2-------3-------4
```

So the final sequence is J1-J4-J2-J3 and the total profit is 110.