7.05.20

Test

DAA Revision -2

Q.1.) Write Prim's Algorithm

Q 2.) Write Kruskal's Algorithm

Q 3.) Write the algorithm for fractical Knapsack's Problem.

A1.) Prims Algorithm for MST

Step 1: Obtain a graph G with V vertices and E edged with w_i weights for each edge $e \in E$.

Step 2: Select a radom vertex v_i next select an adjacent vertex v_j such that the the weight (for the edge e_{uv}) between the vertices u & v is minimum.

Step 3 : Consider another vertex x, such that x is adjacent either to u or to v but not to both, and the weight to x must be smallest among the avaiable choices.

~~Step 4 :~~

Step 4 : Repeat step 2 & 3 keeping note that no loops are formed.

Step 5: Stop the iteration if all the vertices are ~~exact~~ exhausted.

A2.) Kruskal's Algorithm

Step 1: Obtain the graph G with vertex set V and edge set E.

Step 2: Sort the edges according to their weights in the ascending order.

Step 3: Select the edge with minimum weight an create a new sub tree.

Step 4: Again select a new edge, this time 4 cases may occur.

(i) Both vertices of the selected edge may lie in a existing sub tree – discard the edge.

(ii) One of the vertex of the selected edge may line in an existing sub tree – add edge that sub tree

(iii) One edge may lie in one sub tree say T_1 and other may lie in edge T_2 – merge T_1 & T_2

(iv) No ~~end~~ end vertices lie in any sub tree – create a new sub tree.

A3.) Knapsack Algorithm (Greedy & Fractional)

Step 1 : current knapsack load is set to zero
load ← 0 , and counter is initialized to 1.

Step 2 : Repeat the following steps while the
current load < Maximum load.

Step 3 : if w_i + load <= MaxLoad(W)
and i <= no. of items goto step 4
else goto step 5.

Step 4 : take the whole of i^{th} item.

Step 5 : Take only the possible fraction of i^{th} item.
ie load += (W - load).

$$profit += \frac{W - load}{w_i} \times \text{profit of } i^{th} \text{ item}$$

Step 6 : Increment counter i ← i+1

Step 7 : Stop.