**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: B.Tech. Sem V

**Course: Design and Analysis of Algorithms**

**List of Experiments**

w.e.f. 1st Jul 2020

**Faculty:** Abhay Kolhe.

LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.06**

**A.1 Aim:**

Implementation of Greedy Technique Algorithm Design.

Write a program to implement Kruskal’s Algorithm for finding Minimum Spanning Tree (MST). .

**A.2 Prerequisite:**

1. Concepts of Greedy Technique of algorithm design.

2. Knowledge of Graph Handling.

3. Knowledge of Minimum Spanning Tree.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Design & implement a solution using Greedy Technique.
2. Identify different problems that can be solved by using Greedy Technique.
3. Identify applications of Minimum Spanning Tree.

**A.4 Theory:**

**A.4.1.**

**General Concept of Greedy Technique of Algorithm Design**

* An optimization problem is one in which you want to find, not just *a* solution, but the *best* solution
* A “greedy algorithm” sometimes works well for optimization problems
* A greedy algorithm works in phases. At each phase:
  + You take the best you can get right now, without regard for future consequences
  + You hope that by choosing a *local* optimum at each step, you will end up at a *global* optimum

**A Simple Example**

* **Problem:** Pick k numbers out of n numbers such that the sum of these k numbers is the **largest.**
* **Algorithm:**

FOR i = 1 to k

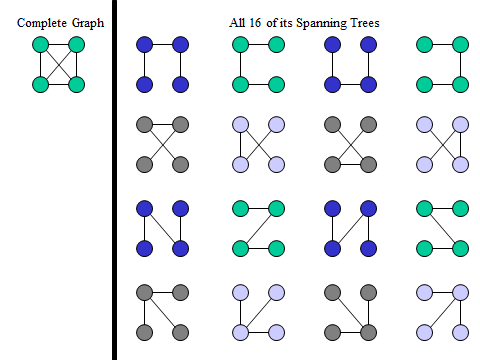
pick out the largest number and

delete this number from the input.

ENDFOR

**A spanning tree** of a graph is just a subgraph that contains all the vertices and is a tree.

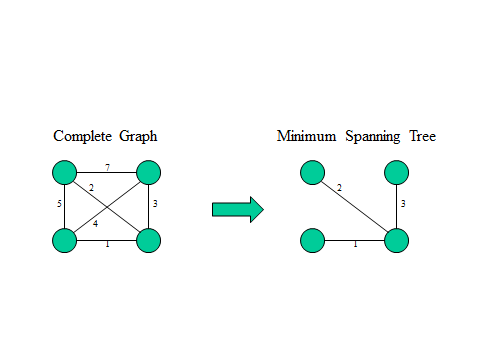
A graph may have many spanning trees**.** Different possible spanning trees for a given graph are as shown in Fig. 1.



**Figure 1: Graph & Its Spanning Trees**

**Minimum Spanning Trees**

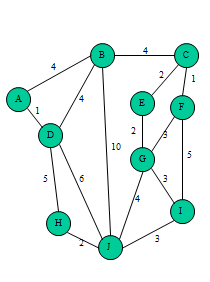
The Minimum Spanning Tree for a given graph is the Spanning Tree of minimum cost for that graph.

****

**Figure 2: A graph & Its spanning tree.**

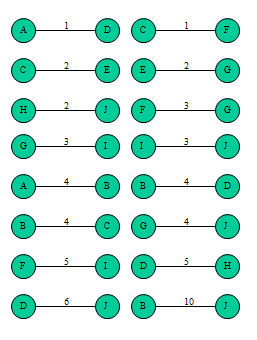
**Kruskal's Algorithm**

This algorithm creates a forest of trees. Initially the forest consists of n single node trees (and no edges). At each step, we add one edge (the cheapest one) so that it joins two trees together. If it were to form a cycle, it would simply link two nodes that were already part of a single connected tree, so that this edge would not be needed.

****

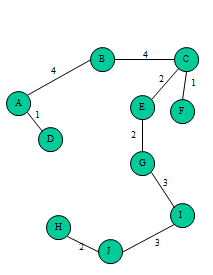
**Figure 3: Complete Graph**

Find all the edges with their corresponding weight and sort them as shown in Fig. 4.

****

**Figure 4: Sorted Edges**

**The resultant minimum spanning tree is as shown in Fig. 5, which has the minimum cost as 22.**

****

**Figure 5: Minimum Spanning Tree**

**A.5 Procedure/Algorithm:**

**A.5.1:**

**Kruskal’s Algorithm**

*1.* Sort the edges *E* in non-decreasing weight2. *T* ← ∅   
3. For each *v* ∈*V* create a set.4. repeat   
5. Select next {*u,v*}∈ *E*, in order   
6. *ucomp* ← *find* (*u*)   
7. *vcomp* ← *find* (*v*)   
8. if *ucomp ≠ vcomp* then  
8. add edge (*u,v*) to *T*   
9. *union* ( *ucomp,vcomp* )  
10.until *T* contains *|V* | - 1 edges  
11. returntree *T*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No. | Name: |
| Class : | Batch : |
| Date of Experiment: | Date of Submission |
| Grade : | Time of Submission: |
| Date of Grading: |  |

**B.1 Software Code written by student:**

***(Paste your c/c++ code completed during the 2 hours of practical in the lab here)***

**B.2 Input and Output:**

***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

**Input Data:**

**Output Data:**

**B.3 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

**B.4 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

**B.5 Question of Curiosity**

***(To be answered by student based on the practical performed and learning/observations)***

Q.1 Identify & discuss in detail the real life applications Kruskal’s Algorithm.

Q.2 Compare the performance of Kruskal’s and Prim’s algorithm.

Q.3 Discuss the time and space complexity of Kruskal’s algorithm.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*