NCKU Programming Contest Training Course
MST
2013/04/10

Pinchieh Huang (free999)

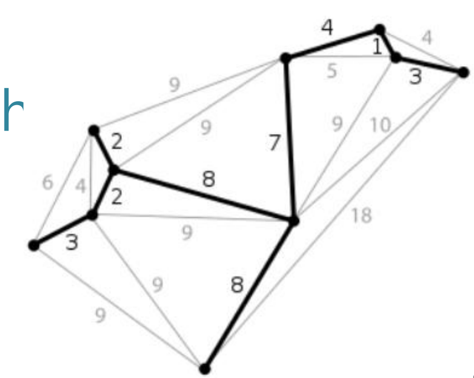
Pinchieh.huang@gmail.com

http://myweb.ncku.edu.tw/~p76014143/20130410_MST.rar

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

MST

- Spanning Tree
  - a spanning tree is a minimum connected graph of G
  - a spanning tree is a tree that is composed of all vertices and some edges of G
  - can also be defined as the maximum edge set without cycle
  - can also be defined as the minimum edge set that connect all vertices
- Minimum Spanning Tree
  - a spanning tree with the smallest weigh



MST

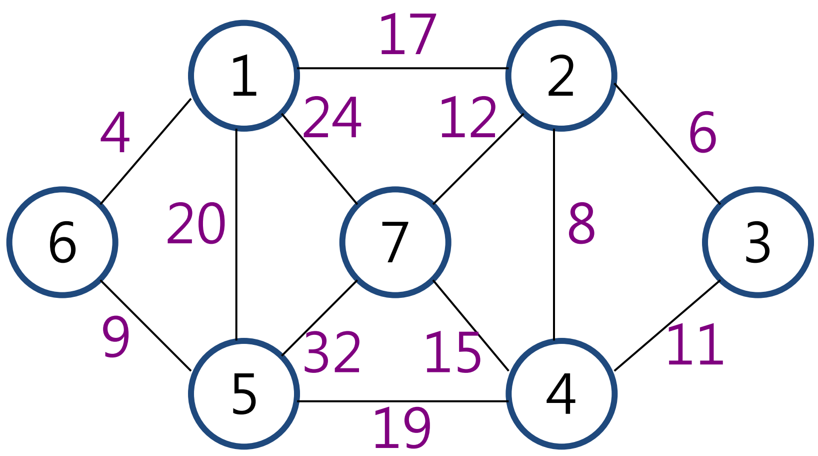
- Algorithm
  - Kruskal’s algorithm
    - greedy based
  - Prim’s algorithm
    - relaxation based

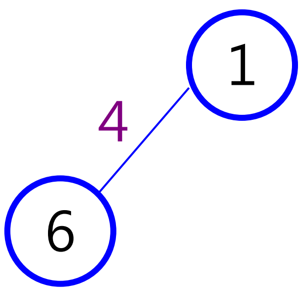
MST

- Kruskal’ s Algorithm
  - Kruskal’ s algorithm is an algorithm in graph theory that find a minimum weight spanning in a connected weighted graph. This mean it finds a subset of edge that forms a tree which includes all the vertices in the graph, where the total weight of all edges in the tree are minimum.
  - if the original graph is disconnected, this algorithm forms a minimum spanning forest
- Main idea
  - sort all the edge by increasing weight
  - iteratively add the minimum weight edge and check if there is a cycle when including the two end vertices of this edge

MST

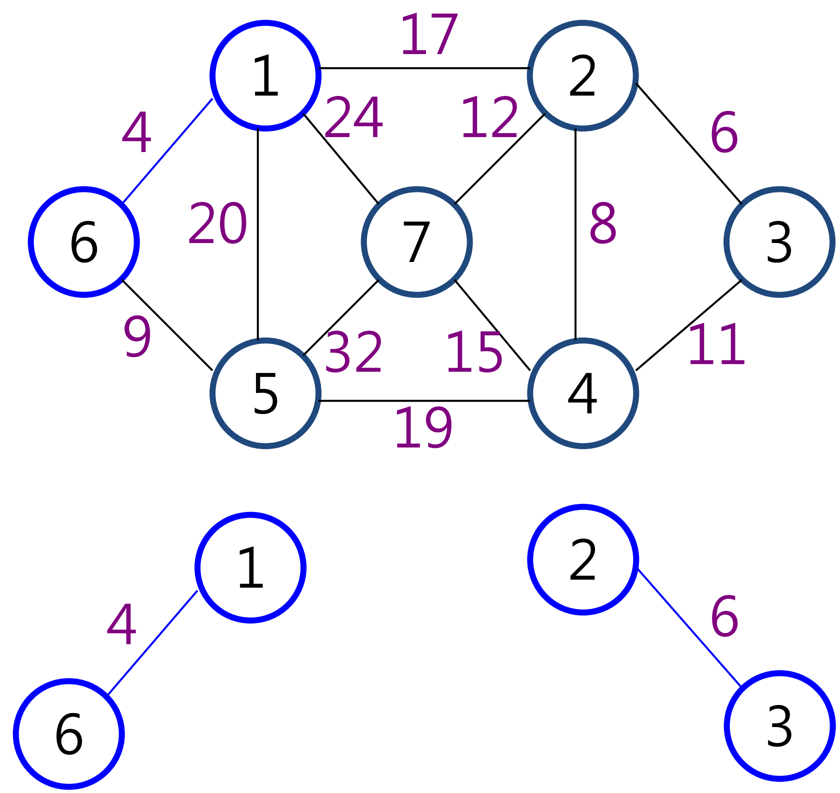
- Initial weighted connected graph
- sort the edge by increasing weight
- pick the minimum (1, 6)





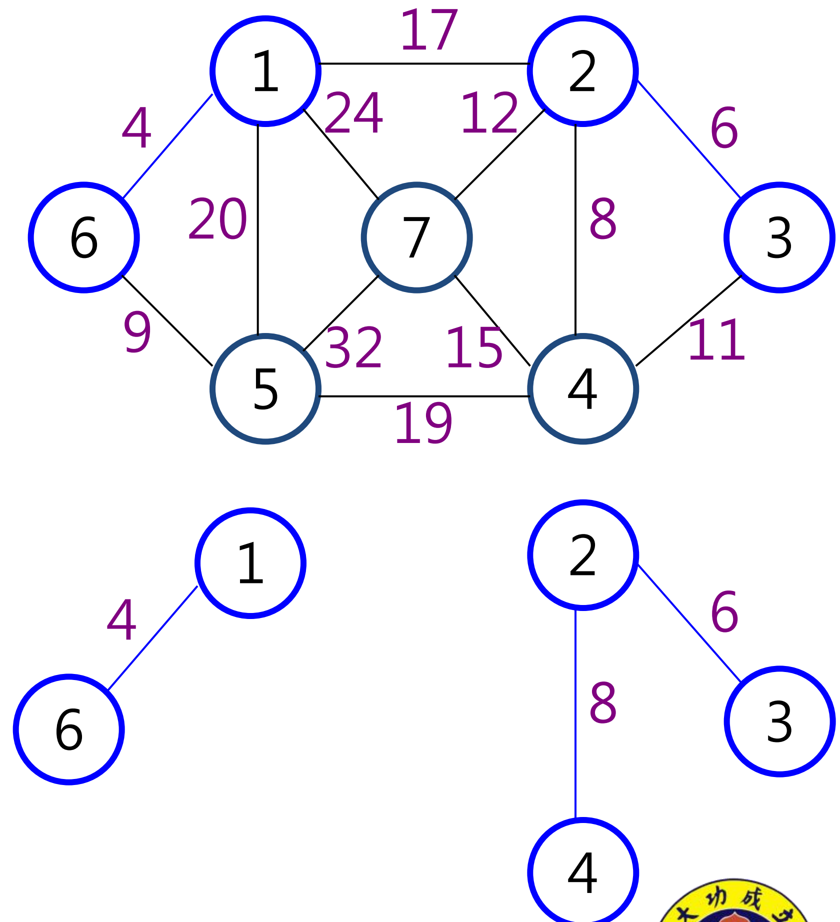
MST

- pick the minimum (2, 3)



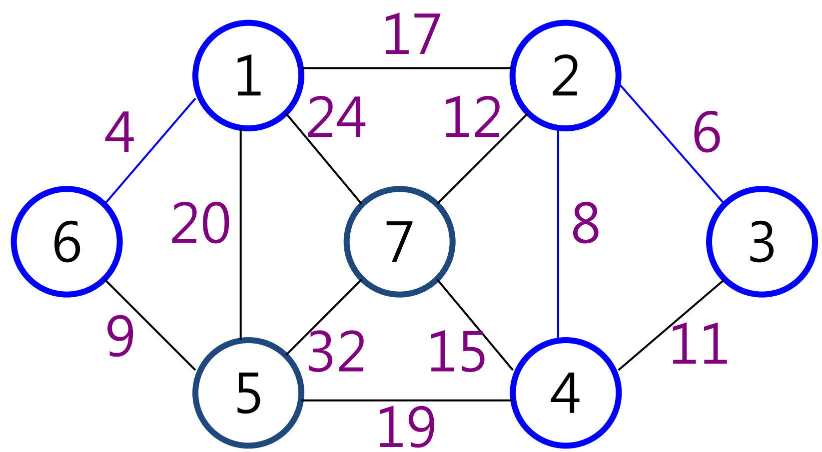
MST

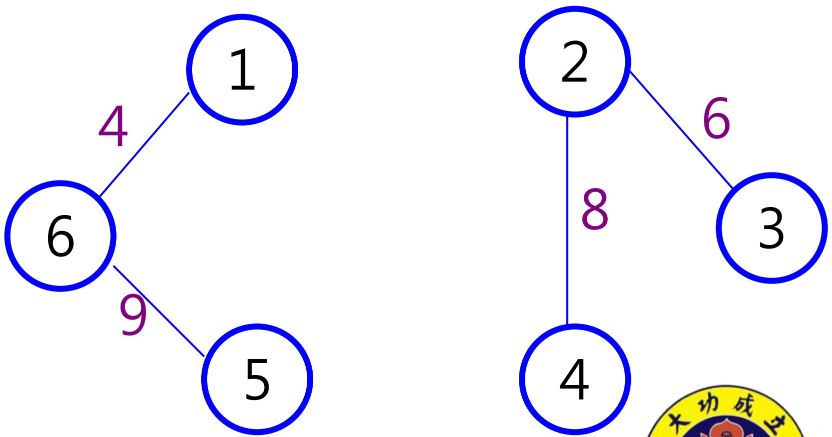
- pick the minimum (2, 4)



MST

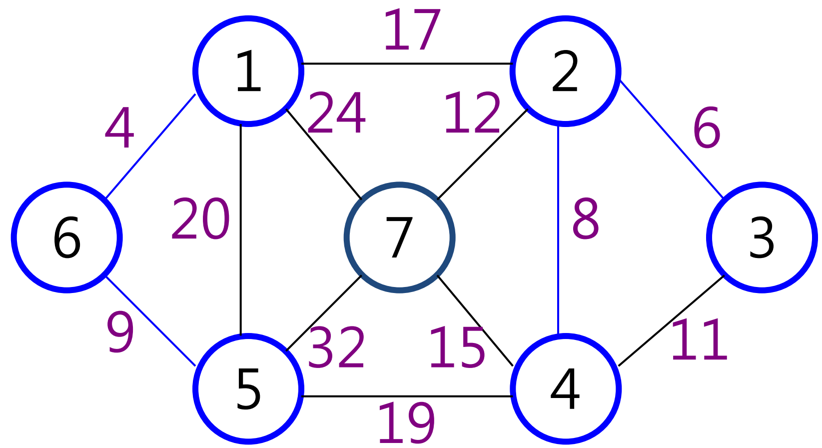
- pick the minimum (5, 6)

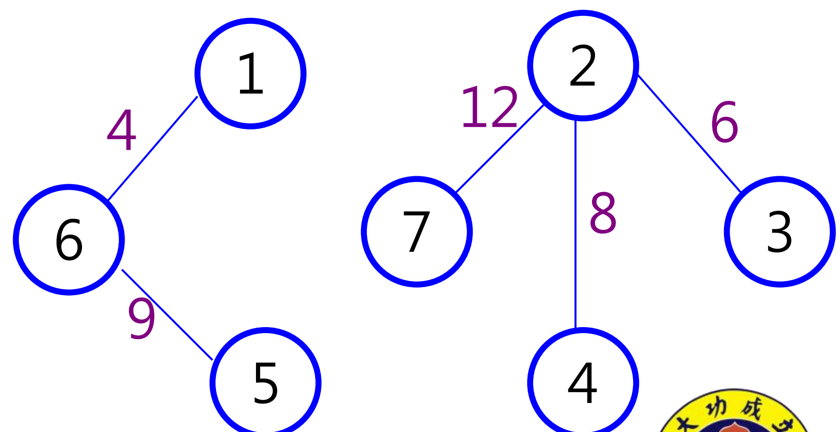




MST

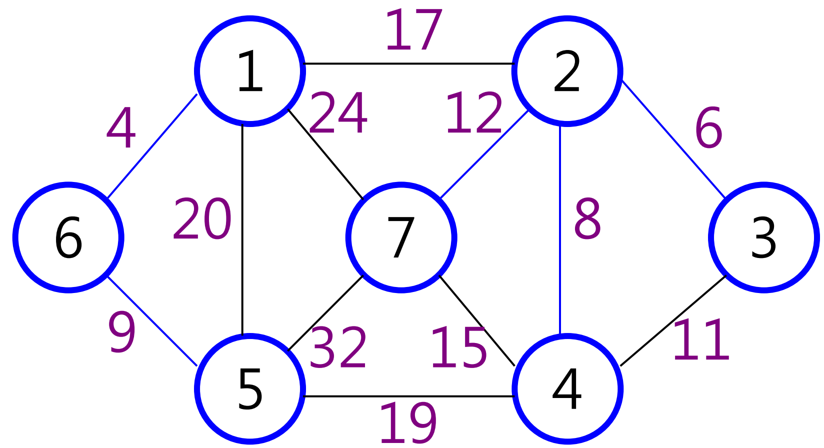
- pick the minimum (3, 4) which forms a cycle
- pick (2, 7)

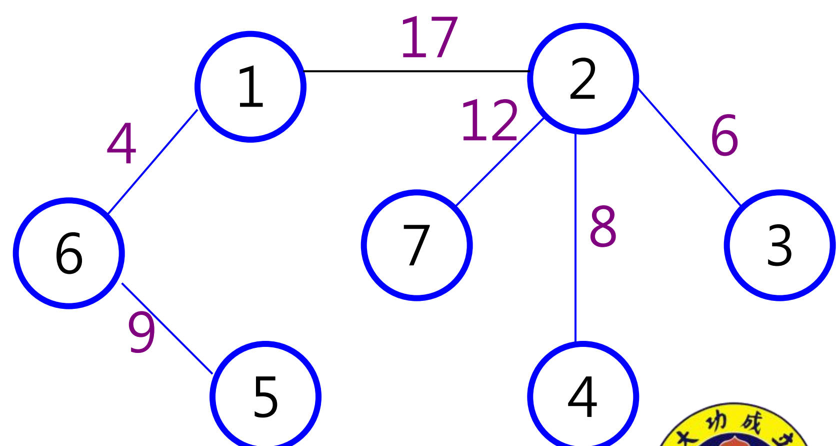




MST

- pick the minimum (4, 7) which forms a cycle
- pick (2, 1)





MST

- Kruskal’s Algorithm
  - Sort edge
  - Check cycle (disjoint set)

Example

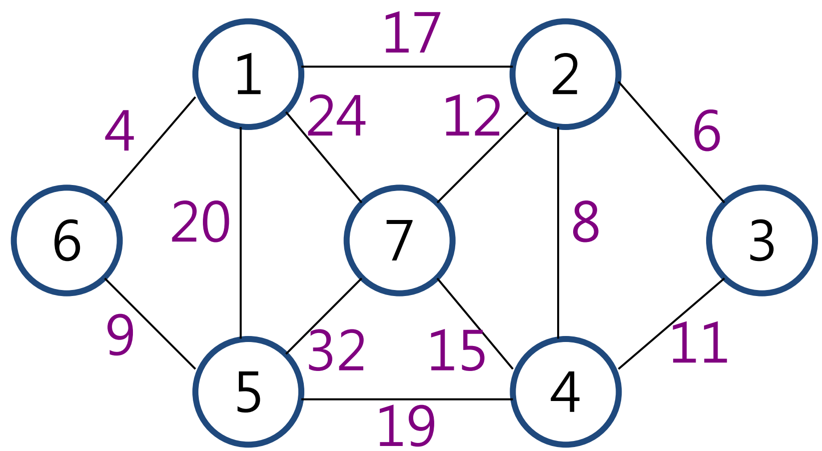
- ICPC 5015

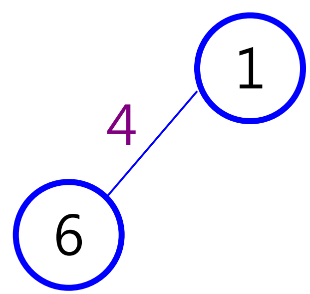
MST

- Prim's Algorithm
  - vertex based expansion

MST

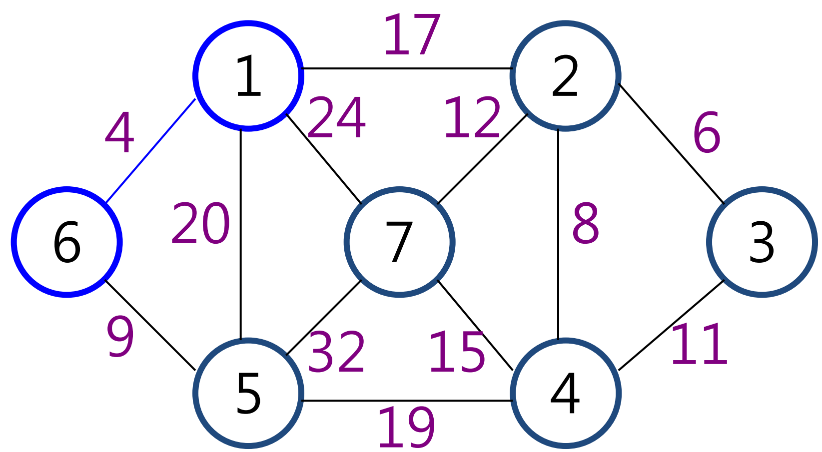
- 若以Prim's algorithm來找最小成本擴展樹，其過程如下：
- V= {1, 2, 3, 4, 5, 6, 7}，U ={1}。
- 從V-U = {2, 3, 4, 5, 6, 7}中找一頂點，與U = {1}頂點能形成最小成本的邊；發現是頂點6，然後加此頂點於U中，U = {1, 6}。

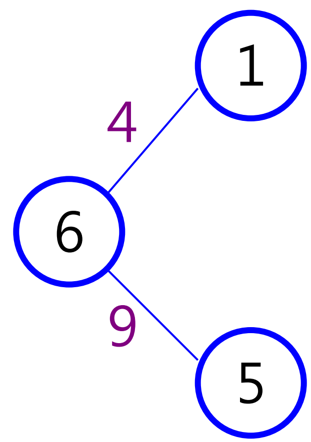




MST

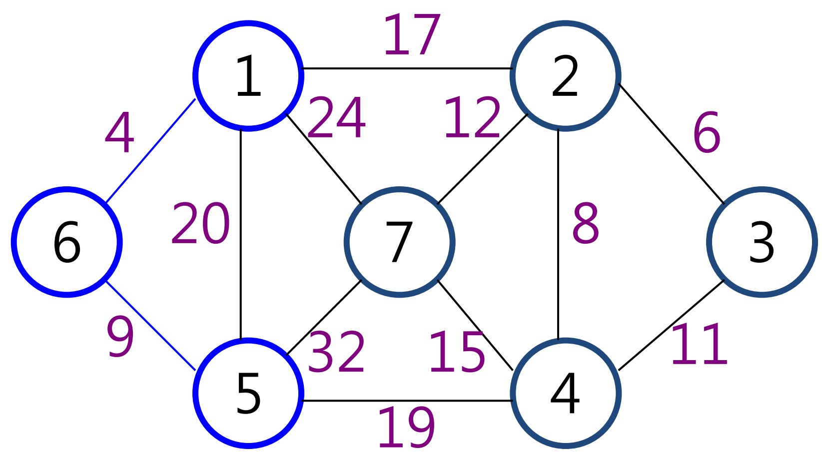
- 此時V-U = {2, 3, 4, 5, 7}，從這些頂點找一頂點，與U = {1, 6}頂點能形成最小成本的邊，答案是頂點5，因為其成本或距離為9；加此頂點於U中，U = {1, 5, 6}，V-U = {2, 3, 4, 7}。

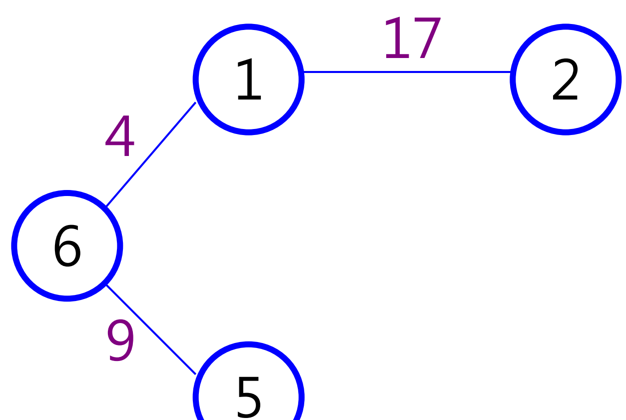




MST

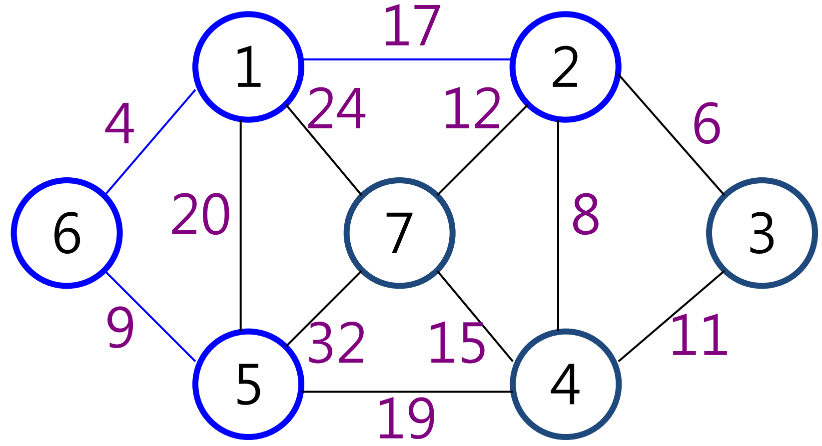
- 以同樣方法找到一頂點2，能與V中的頂點1形成最小的邊，加此頂點於U中，U = {1, 2, 5, 6}，V-U = {3, 4, 7}

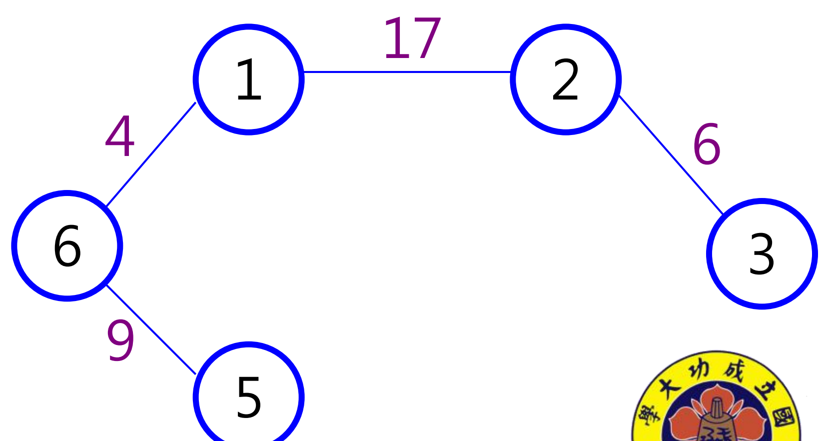




MST

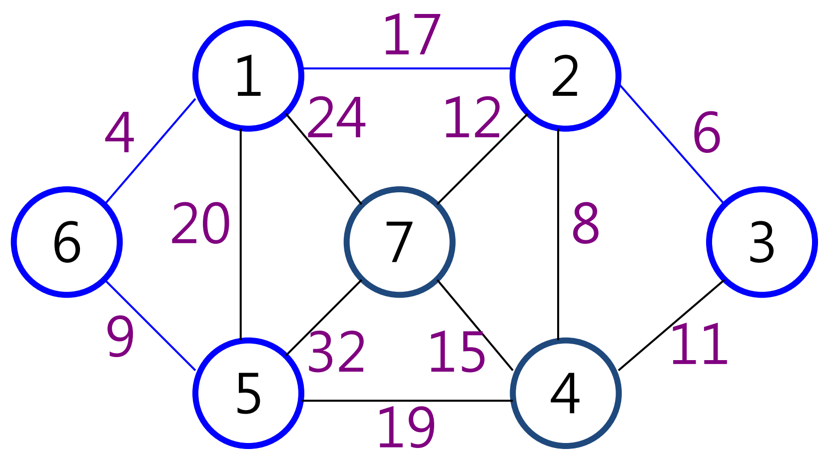
- 同樣方法將頂點3加入U中，U ={1, 2, 3, 5, 6}，V-U = {4, 7}。

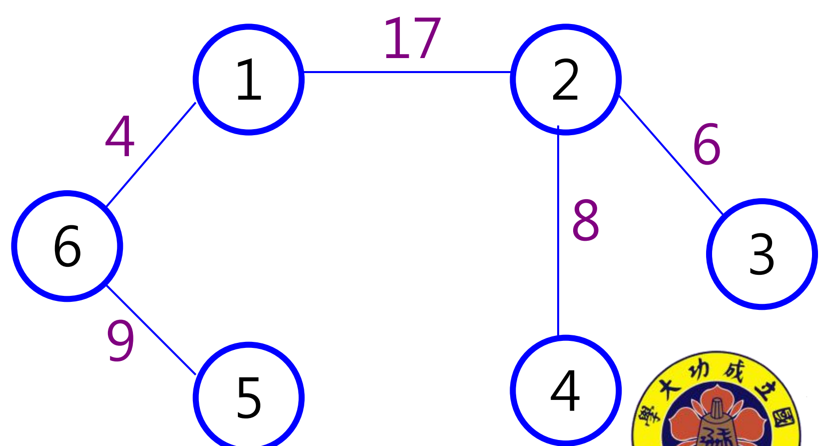




MST

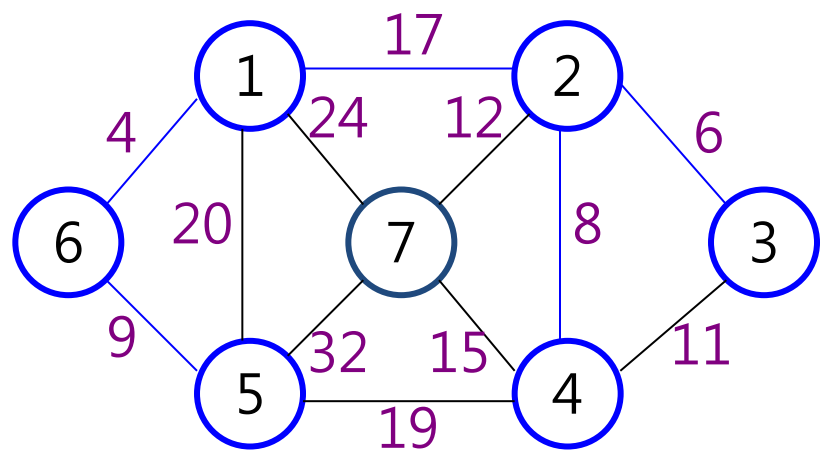
- 以同樣的方法將頂點4加入U中，U = {1, 2, 3, 4, 5, 6}，V-U = {7}。

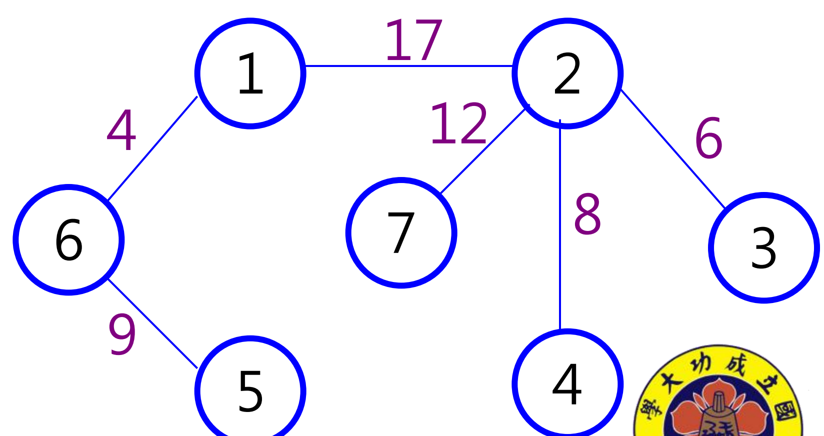




MST

- 將頂點7加入U中，U = {1, 2, 3, 4, 5, 6, 7}，V-U =∮，V = U，此時的圖形就是最小成本擴展樹。





MST

- Prim’ s Algorithm
  - 1. 所有節點設為未拜訪過
  - 2. 令 d[i] 為到 節點 i 的目前距離，起使皆設為INF
  - 3. 每次都去找未拜訪過的節點 i ，而且d[i] 最小
  - 4. 找完後要更新未拜訪過的節點距離 d[j] . if 找到的節點 i 到節點 j 的距離小於d[j] 則要更新d[j]

Example

POJ 2485 Highways

Description

The island nation of Flatopia is perfectly flat. Unfortunately, Flatopia has no public highways. So the traffic is difficult in Flatopia. The Flatopian government is aware of this problem. They're planning to build some highways so that it will be possible to drive between any pair of towns without leaving the highway system.

Flatopian towns are numbered from 1 to N. Each highway connects exactly two towns. All highways follow straight lines. All highways can be used in both directions. Highways can freely cross each other, but a driver can only switch between highways at a town that is located at the end of both highways.

The Flatopian government wants to minimize the length of the longest highway to be built. However, they want to guarantee that every town is highway-reachable from every other town

Example

POJ 2485 Highways

Input

The first line of input is an integer T, which tells how many test cases followed.
The first line of each case is an integer N ($3 <= N <= 500$), which is the number of villages. Then come N lines, the i-th of which contains N integers, and the j-th of these N integers is the distance (the distance should be an integer within [1, 65536]) between village i and village j. There is an empty line after each test case.

Output

For each test case, you should output a line contains an integer, which is the length of the longest road to be built such that all the villages are connected, and this value is minimum.

Example

POJ 2485 Highways

Sample Input

1

3

0 990 692

990 0 179

692 179 0

Sample Output

692

Homework

- UVA (total 40 problems)
  - 10034, 10147, 10397, 10600, 10842, 908, 10369, 10807
- POJ
  - 2485, 1861, 2395, 2377, 2421, 1679, 1751, 1354, 1258, 1251, 3625, 1789

Thank You For Attention!