NCKU Programming Contest Training Course
Minimum Spanning Tree
2017/04/26

Chien-Wen Chen

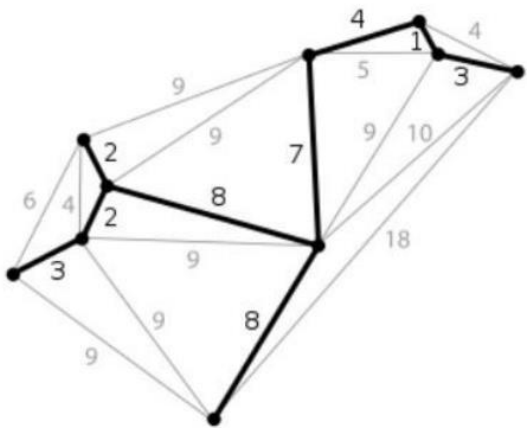
ai281918@gmail.com

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

made by ai281918

MST

- SpanningTree
  - a spanning tree is a tree that is composed of all vertices in the graph
  - can also be defined as the maximum edge set without cycle
  - can also be defined as the minimum edge set that connect all vertices
  - a graph may have many spanning trees
- Minimum SpanningTree
  - a spanning tree with the smallest weight

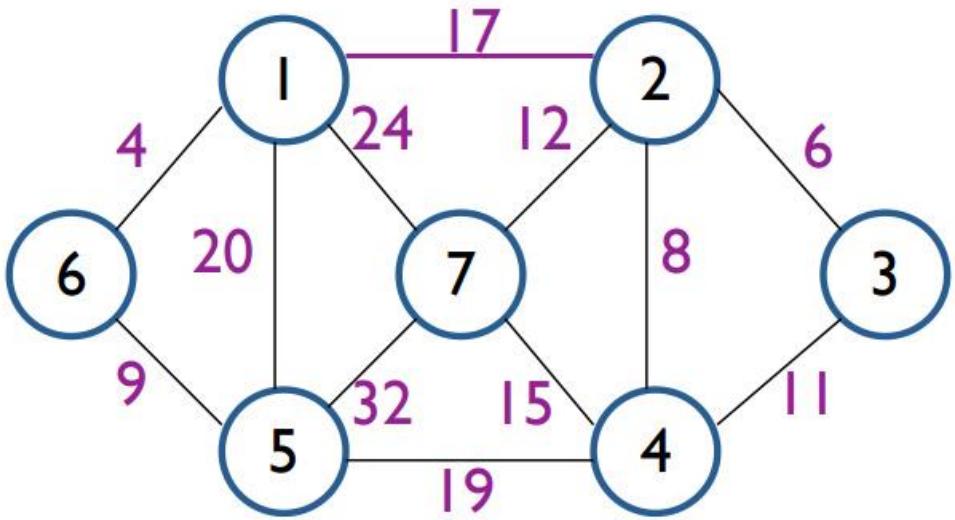


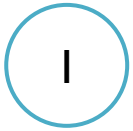
MST

- Algorithm
  - Prim's algorithm
    - relaxation based
  - Kruskal's algorithm
    - greedy based
    - disjoint-set

Prim

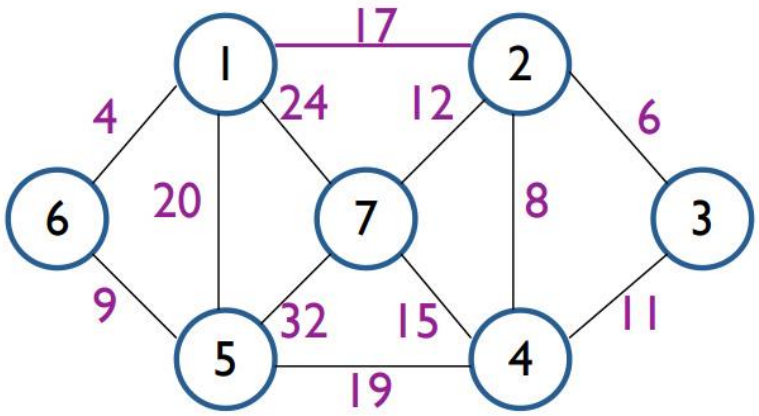
- We have vertex V = {1, 2, 3, 4, 5, 6 ,7} and initial tree T = {1}
- For each step, find a vertex which have minimum cost to add

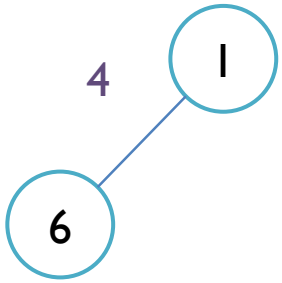




Prim

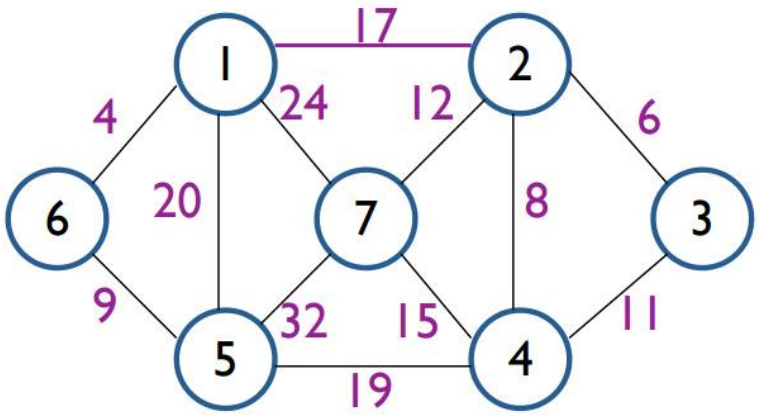
- Step 1 : add 6

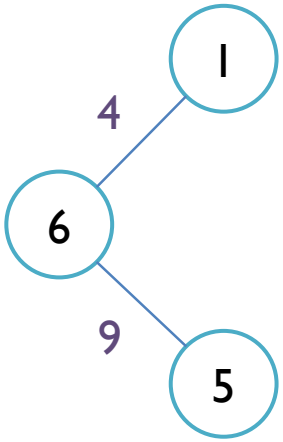




Prim

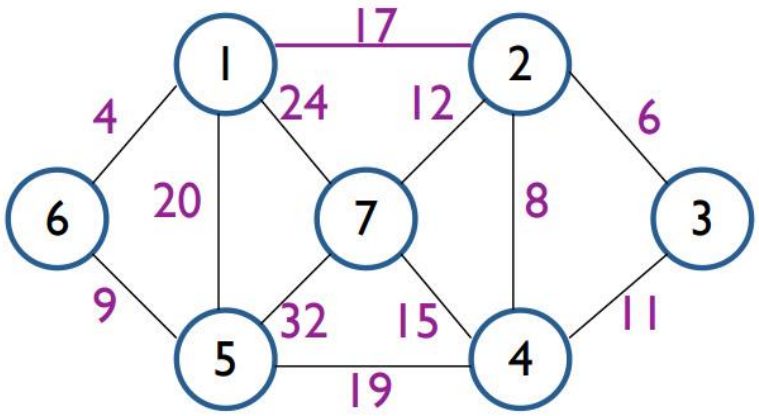
- Step 2 : add 5

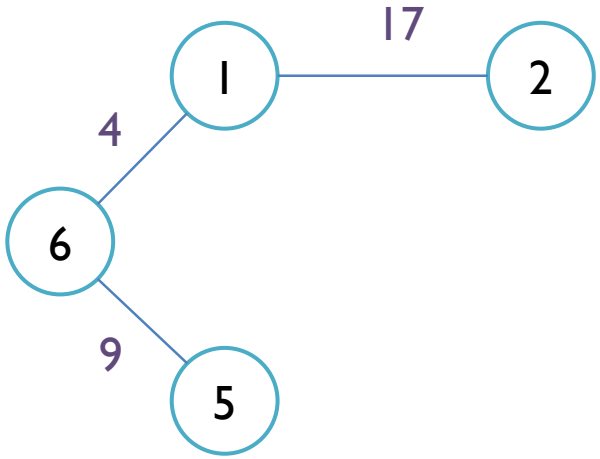




Prim

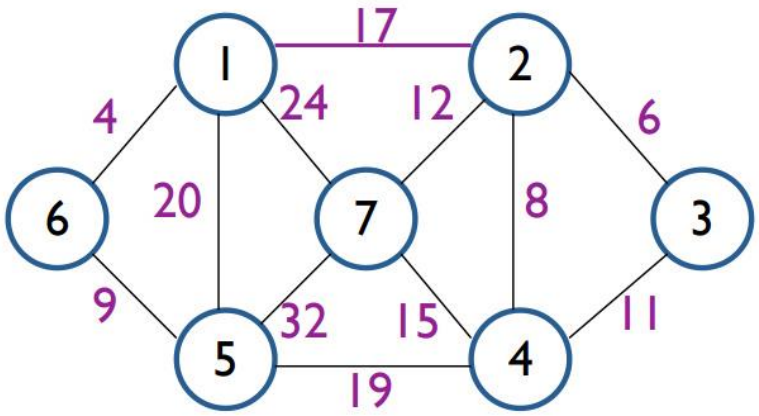
- Step 3 : add 2

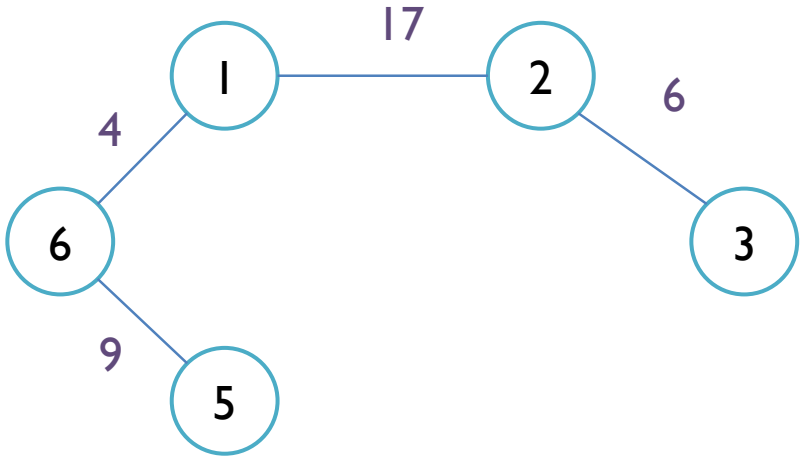




Prim

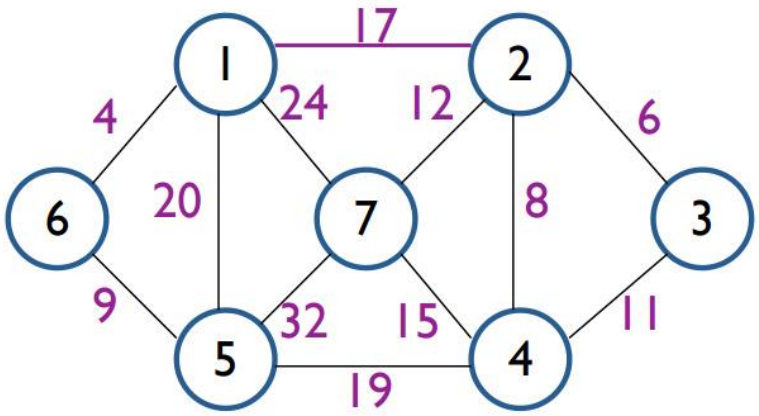
- Step 4 : add 3

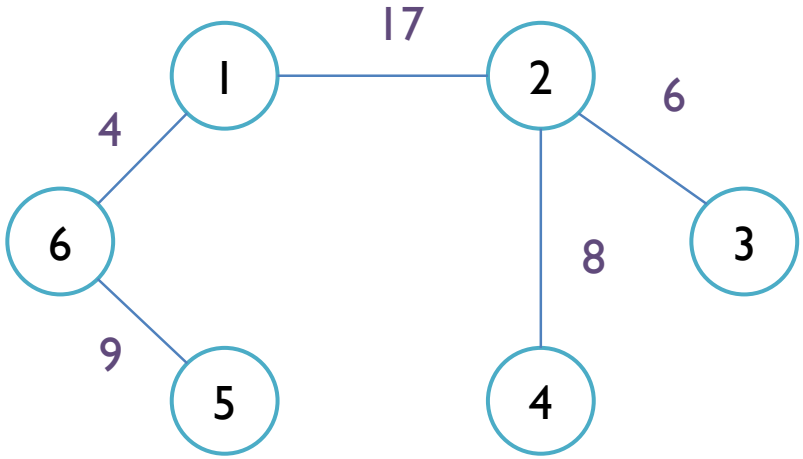




Prim

- Step 5 : add 4

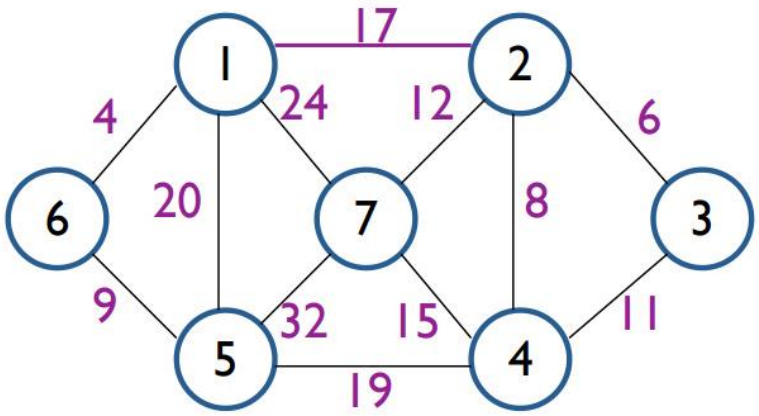


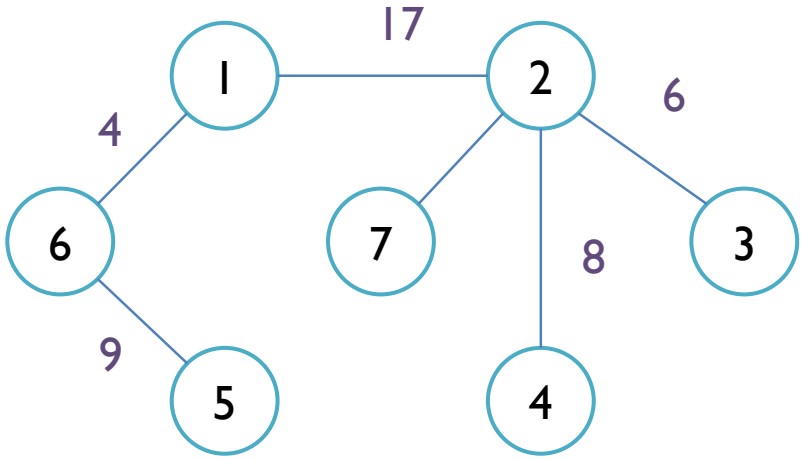


made by ai281918

Prim

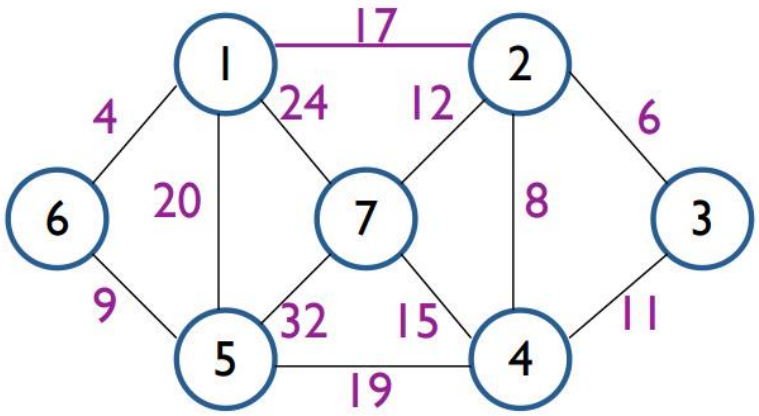
- Step 6 : add 7

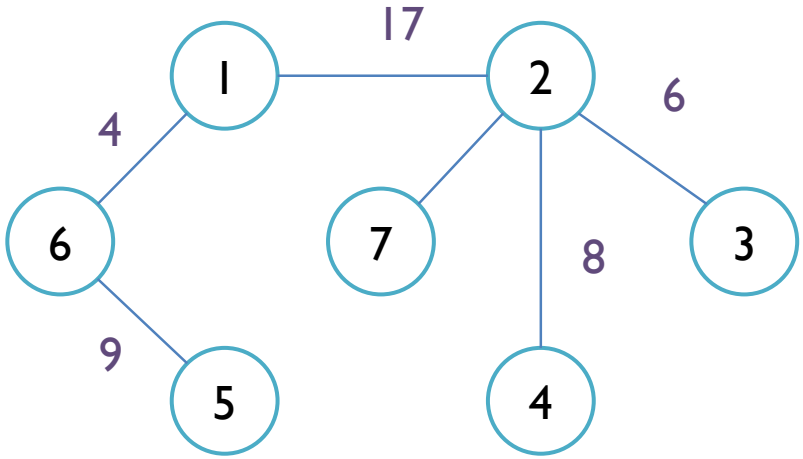




Prim

- Time complexity $O(V^2)$



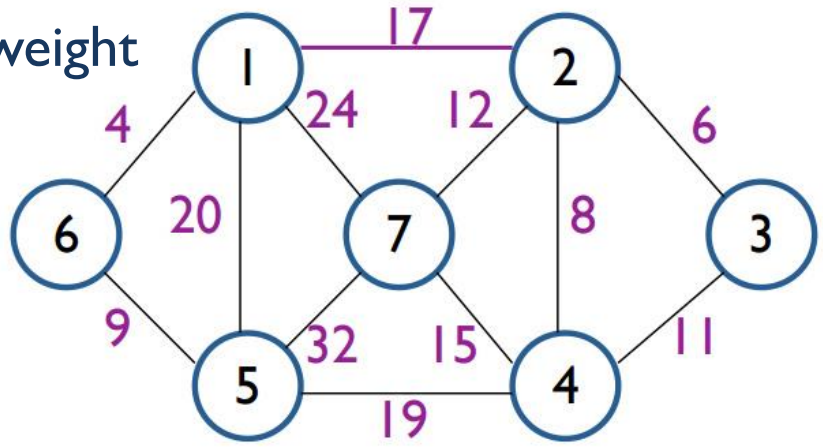


Kruskal

- Sort the edges by increasing weight
- Pick the minimun edge and Check cycle (disjoint set)

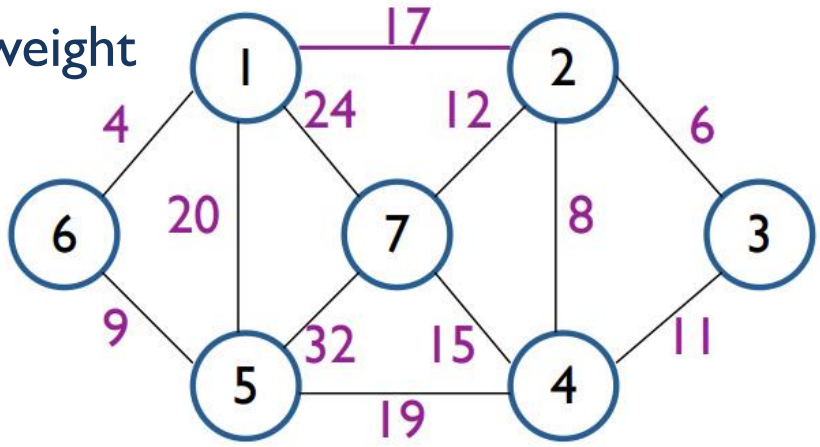
Kruskal

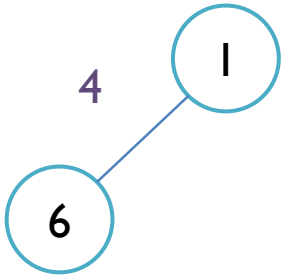
- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- (3,4) :11
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32



Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- (3,4) :11
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32

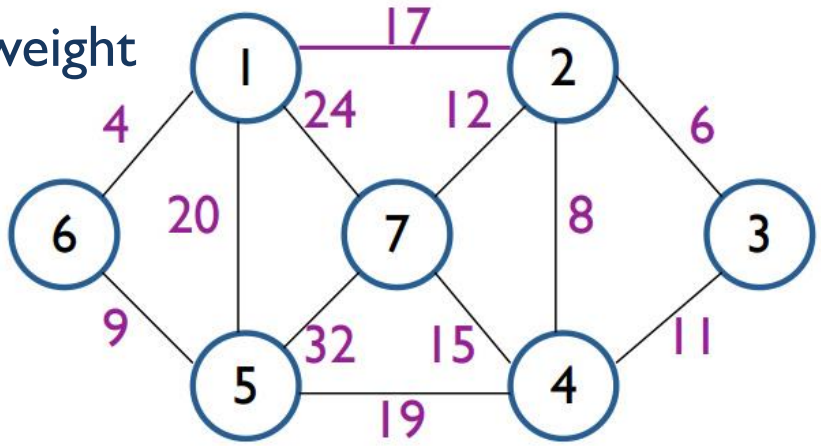




made by ai281918

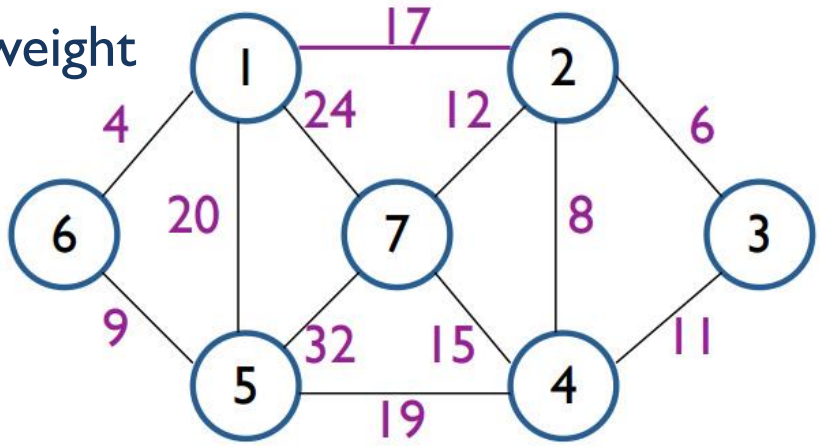
Kruskal

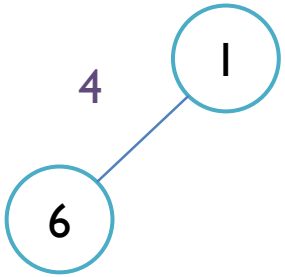
- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- (3,4) :11
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32

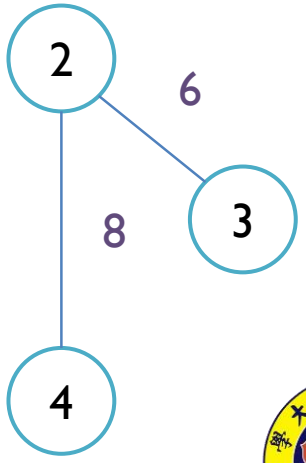


Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- (3,4) :11
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32

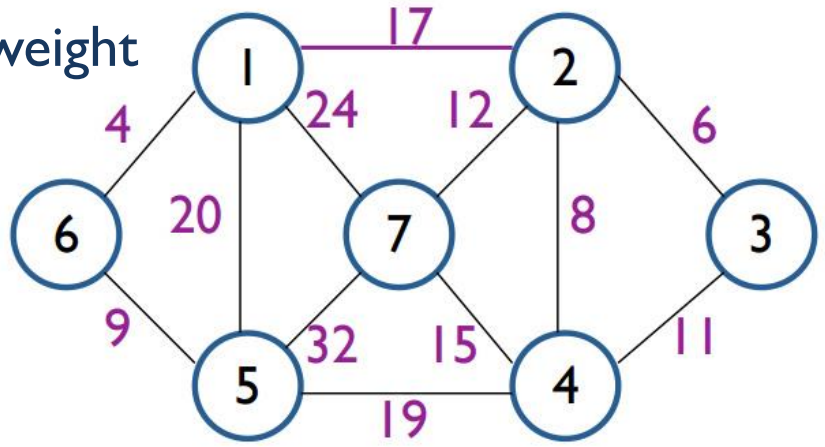


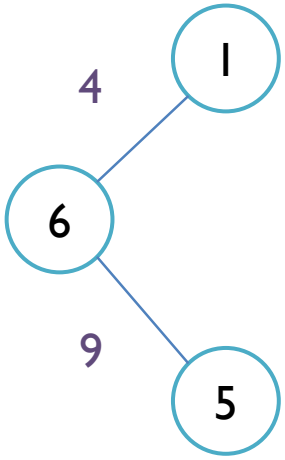


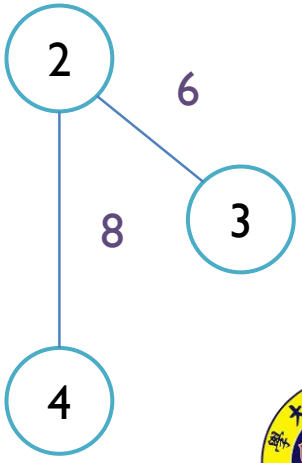


Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- (3,4) :11
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32



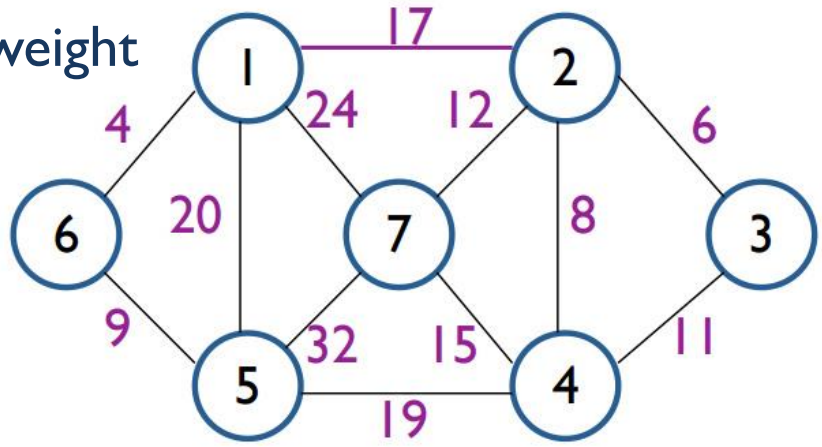


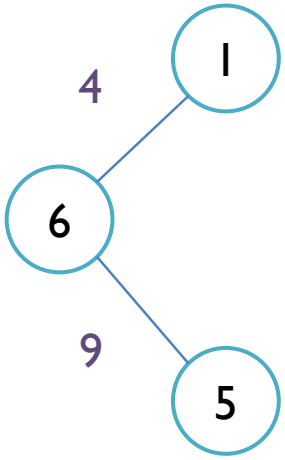


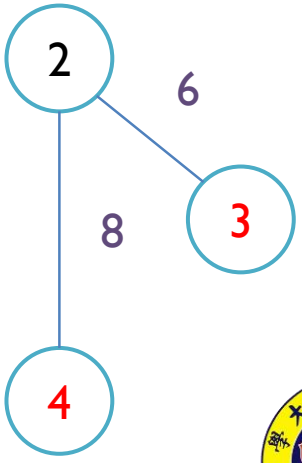
made by ai281918

Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- ~~(3,4) :11~~
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32



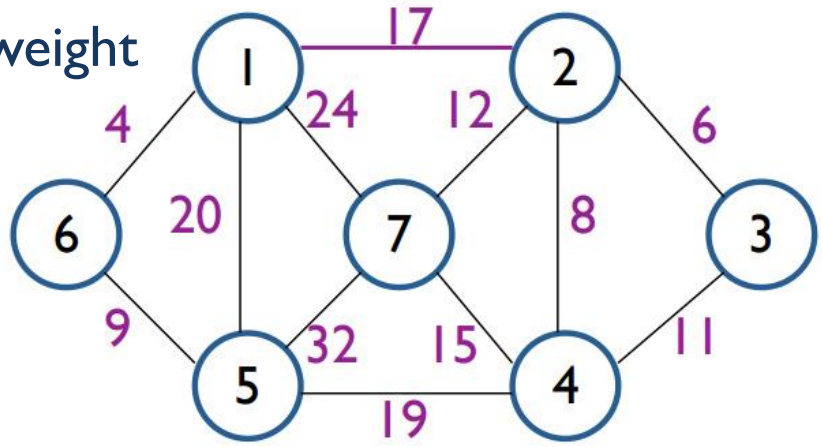


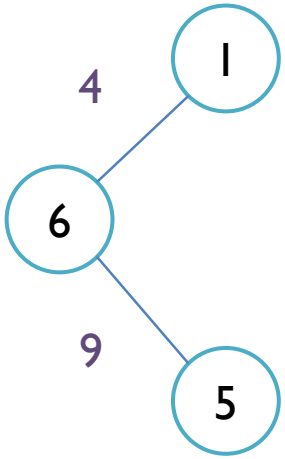


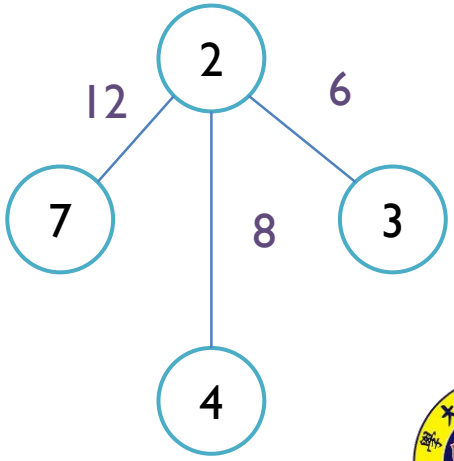
made by ai281918

Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- ~~(3,4) :11~~
- (2,7) :12
- (4,7) :15
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32



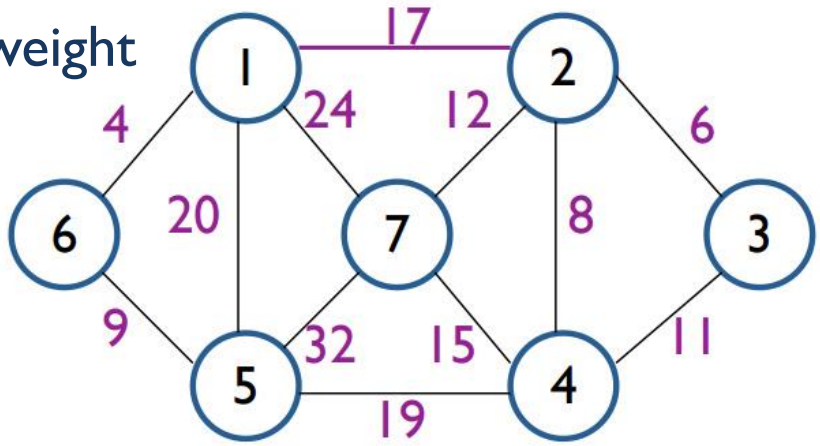


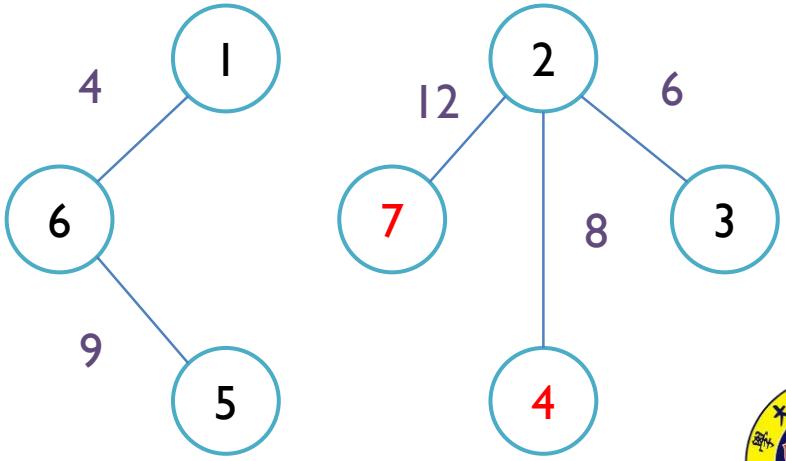


made by ai281918

Kruskal

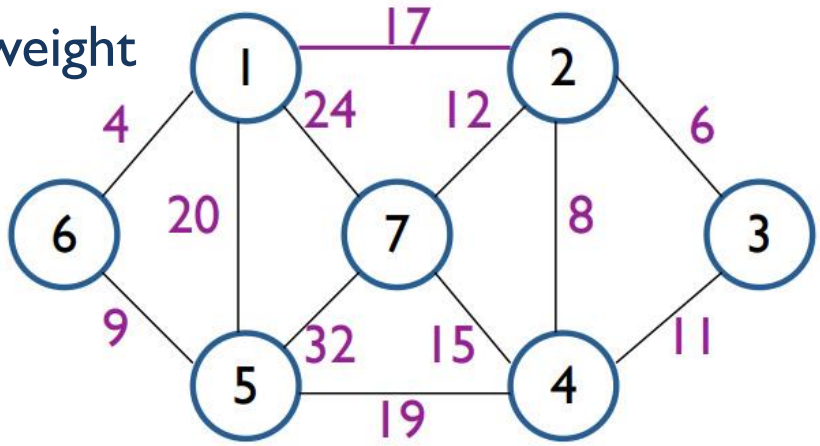
- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- ~~(3,4) :11~~
- (2,7) :12
- ~~(4,7) :15~~
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32

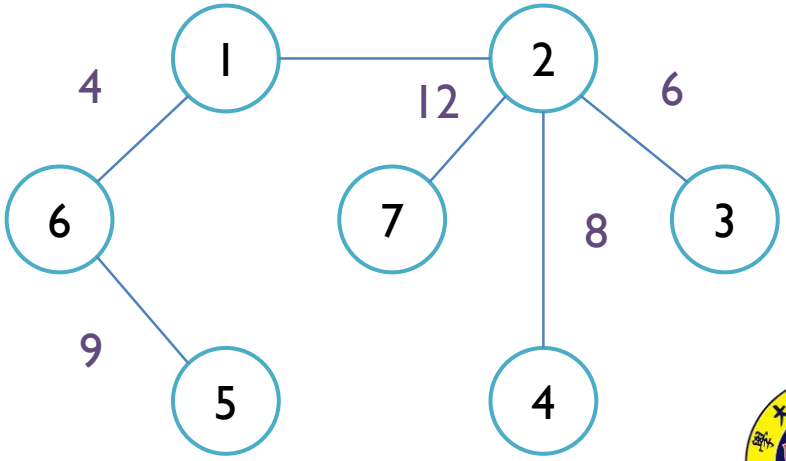




Kruskal

- Sort the edges by increasing weight
- (1,6) :4
- (2,3) :6
- (2,4) :8
- (5,6) :9
- ~~(3,4) :11~~
- (2,7) :12
- ~~(4,7) :15~~
- (1,2) :17
- (4,5) :19
- (1,5) :20
- (1,7) :24
- (5,7) :32





made by ai281918

Kruskal

- Psuedocode

```
totalcost← 0
sort the edges into non-decreasing order by weight
for each edge (u, v) ∈ E, taken in non-decreasing order
        do if FIND-SET (u) ≠ FIND-SET (v)
                then UNION (u,v)
                        totalcost← totalcost+ w(u,v)
return totalcost
```

Kruskal

- Time complexity : O(Elog(E)) (sorting edge)

Practice - 1

UVa 10034 - Freckles

Problem Description

In an episode of the Dick Van Dyke show, little Richie connects the freckles on his Dad's back to form a picture of the Liberty Bell. Alas, one of the freckles turns out to be a scar, so his Ripley's engagement falls through.

Consider Dick's back to be a plane with freckles at various (x, y) locations. Your job is to tell Richie how to connect the dots so as to minimize the amount of ink used. Richie connects the dots by drawing straight lines between pairs, possibly lifting the pen between lines. When Richie is done there must be a sequence of connected lines from any freckle to any other freckle.

Practice - 1

UVa 10034 - Freckles

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line contains $0 < n \leq 100$, the number of freckles on Dick's back. For each freckle, a line follows; each following line contains two real numbers indicating the (x, y) coordinates of the freckle.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Your program prints a single real number to two decimal places: the minimum total length of ink lines that can connect all the freckles.

Problem List

- UVa
  - 10034, 10147, 10397, 10600, 10842, 908, 10807, 11710, 11987, 10369, 11597
- POJ
  - 1258,1251,1354,1679,1789,2377,3625,1751,1861,2395,2421,2485,2560
- 門檻：5題
- 第二次修課同學，請從紅字中挑選5題來完成門檻