return dist[z]

Trees

TREES

Tree: undirected graph that is connected and has no cycles

↳ the graph is a tree

↳ the graph is connected and has at most n-1 edges

↳ the graph has no cycles and at least n-1 edges

↳ the graph is connected and minimal (if we remove any edge it becomes unconnected)

↳ the graph has no cycles and is maximal (if we add any edge, it closes a cycle)

↳ for any pair of vertices, there is a unique path connecting them

THE MINIMUM SPANNING TREE PROBLEM

Given a graph with non-negative costs, find a tree with the same vertices and a subset of the edges of the original graph (a spanning tree) of minimum total cost.

Algorithms for solving this: → Kruskal, → Prim

KRUSKAL ALGORITHM

↳ greedy algorithm

Complexity: $O(E \log E + E \log V)$ (sorting edges + processing vertices)

Steps

1.) Sort all edges in ascending order of their weight

2.) i) Pick the smallest edge
ii) Check if the new edge forms a cycle in our spanning tree being formed
iii) If cycle is not formed → include the edge
else → discard the edge

Repeat Step 2 until V-1 edges are included in the MST

The basic algorithm

Input
G - undirected graph with costs

Output:
edges - a collection of edges forming a MST

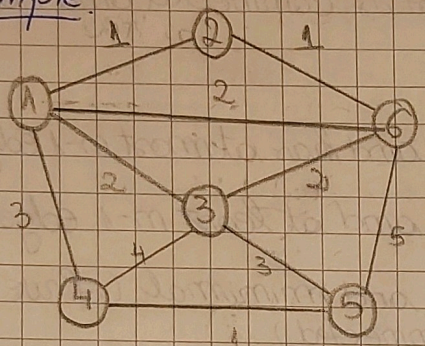
Algorithm:

$e_0, \dots, e_{m-1}$ = list of edges sorted increasing by cost

edges = $\emptyset$

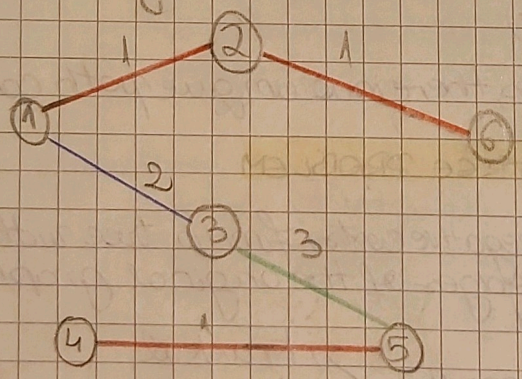
for i from 0 to m-1 do
    if edges $\cup \{e_i\}$ has no cycles then
        edges.add(e_i)

Example



Edges	Cost
(1,2)	1
(2,6)	1
(4,5)	1
(1,6)	2
(1,3)	2
(3,6)	2
(1,4)	3
(3,5)	3
(4,3)	4
(5,6)	5

Building the MST



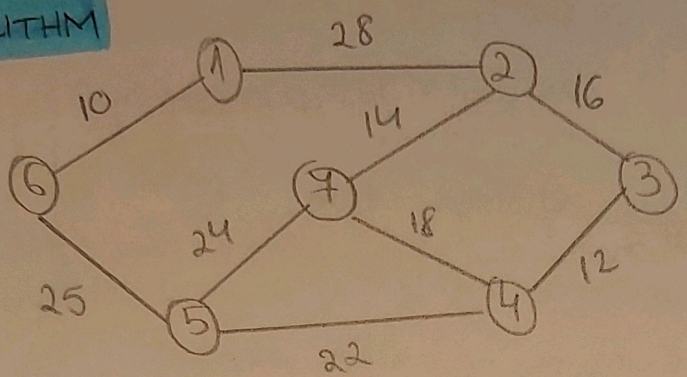
An MST of cost 8

3 edges with cost 1
(1,6) creates a cycle
(3,6) creates a cycle
2 edges with cost 2
3 edges with cost 3
(4,3) creates a cycle
(5,6) creates a cycle

How to test the existence of cycles?

Alternative: keep track of the connected components of edges and notice that a cycle is formed adding a new edge $\Leftrightarrow$ the endpoints of the edge are in the same component

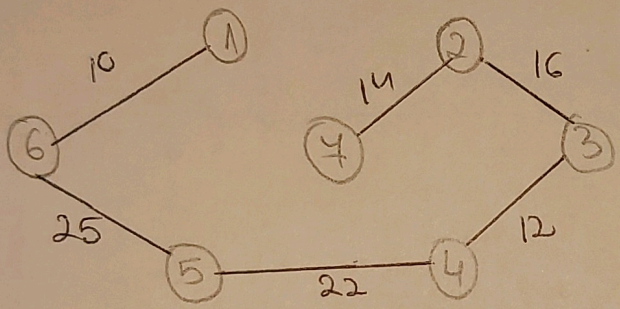
KRUSKAL ALGORITHM



List of sorted edges

Edge	Cost
(1,6)	10
(3,4)	12
(2,7)	14
~~(2,3)~~	16
(7,4)	18
(5,4)	22
(5,7)	24
(5,6)	25
(2,1)	28

MST



Step 1: (1,6) ✓

Step 2: Choose (3,4)

Step 3: Choose (2,7)

Step 4: Choose (2,3)

Step 5: (7,4) is not a valid edge

Step 6: Choose (5,4)

Step 7: (5,7) is not a valid edge

Step 8: Choose (5,6)

6 edges, 7 vertices → 1 component that is a tree