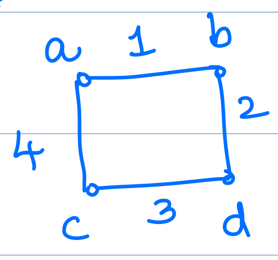
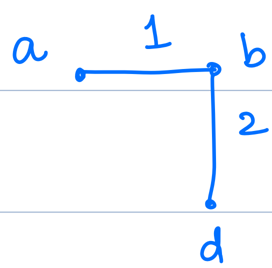
Steiner tree Problem

Given an undirected graph $G=(V,E)$ with non-negative edge weights and a subset of vertices R (called terminals) the Steiner tree Problem is to find a minimum cost tree that contains all terminals (may include additional vertices).

Steiner vertices.

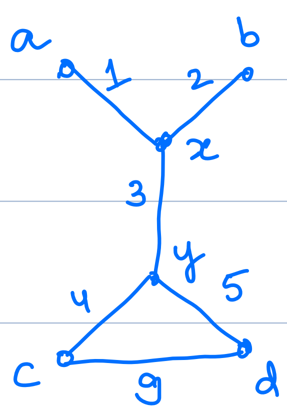
Examples

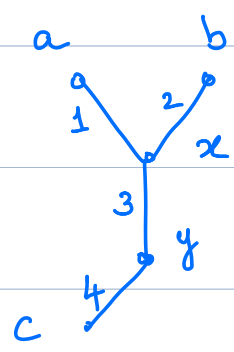
① 



R = Terminals $= \{a, d\}$

Steiner tree of weight 3

② 



R = Terminals $= \{a, b, c\}$

Steiner tree of weight 10

Remarks:

1. If a Steiner tree problem in graphs contains exactly two terminals, it reduces to finding the shortest path.

2. If a Steiner tree problem in graphs contains all vertices as terminals, it reduces to finding the minimum spanning tree (MST)

3. We know that non-negative shortest path and MST problem are solvable in polynomial time. However decision version of Steiner tree problem is NP-Complete.

Metric Steiner tree Problem

It is a Steiner tree Problem in which G is a complete graph and edge costs satisfiy the triangle inequality

[We will work on Metric Steiner tree]

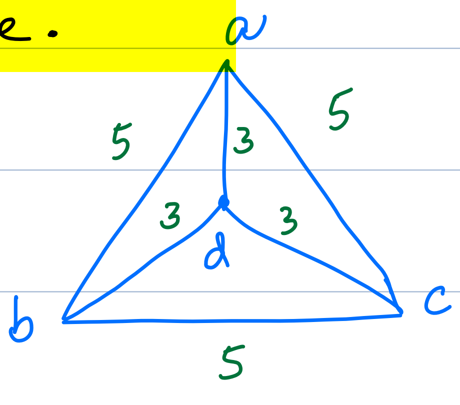
MST-based Algorithm:

let R denote the Set of required vertices.

Clearly a MST on R is a feasible solution.

But MST on R may not give optimal Steiner tree.

Eg



$R = \{a, b, c\}$

Cost of MST on $R = 10$

$OPT = 9$

Theorem: The cost of an MST on R is within 2.OPT.

Proof:- Consider a Steiner tree T of cost OPT. By doubling the edges of T we obtain an Eulerian graph connecting all vertices of R and possibly some Steiner vertices.

Find an Euler tour of this graph.

Clearly the cost this Euler tour is 2.OPT

Next obtain a Hamiltonian cycle on the vertices of R by traversing the Euler tour and "short-cutting" Steiner vertices and previously visited vertices on R.

Because of triangle inequality, the shortcuts do not increase the cost of the tour.

If we delete one edge of this Hamiltonian cycle, we obtain a path that spans R and has cost at most $2OPT$. This path is also a spanning tree on R. Hence, the MST on R has cost at most $2 \cdot OPT$.

Exercise:

1. The hardness of the Steiner tree Problem lies in determining the optimal subset of Steiner vertices that need to be included in the tree. Show this by Proving that if this set is Provided then optimal Steiner tree can be Computed in Polynomial time.