6.7 Minimum Spanning Trees (MSTs)

Definition 19. *Given a weighted (undirected) connected graph G, its* minimum spanning tree (MST) *is a subgraph containing all the vertices with minimum total weight.*

A *spanning tree* is a connected subgraph that covers all the vertices with minimum number of edges. This subgraph cannot contain a cycle, for we can remove an edge and the resulting graph is still connected.

→ cycle 있으면 낭비임.

① cycle 없으면서,

② connected, (undirected)

③ 모든 꼭짓점 cover.

Figure 9: A spanning tree is a kind of skeleton of a graph.

The importance of MST seems obvious; the following are some practical problems that can be explained in terms of MST:

- Connections in electric curcuit with minimum wiring.
- Routing ferries with a minimum cost.
- Computer network applications: e.g., spanning tree protocol.
- Machine learning: e.g., single-link clustering.

Before we discuss how to find minimum spanning trees, let us first talk about the "tree" in the term. A connected graph without a cycle is called a *free tree*, or simply a tree. If we forget the root from a rooted tree, it becomes a free tree. A minimum spanning tree must be a free

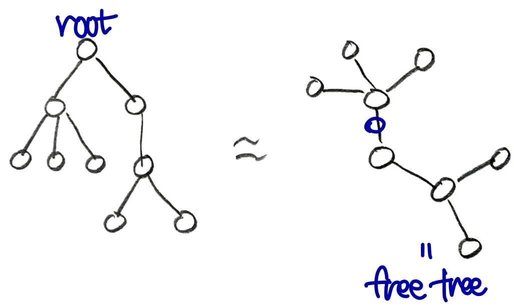

Figure 10: If we forget the root of a tree, we get a free tree. (connected ★ no cycle

꼭짓점 n개 ↕ edge n-1개

tree, because if it contains a cycle we can remove an edge to obtain a subgraph with less weight containing all the vertices.

free

free tree

Useful facts regarding trees (and hence MSTs) When G is a graph with n vertices, the following are equivalent: 꼭짓점 n개 → 선 n-1개 (edge)

(a) G is a free tree. (connected with no cycle)
↳ 있으면 낭비.

(b) G is connected, and if any edge is deleted, then the remaining graph is no longer connected.

(c) If v and v' are distinct vertices, then there is exactly one simple path from v to v'.

(d) G contains no cycle and has $n-1$ edges.

(e) G is connected and has $n-1$ edges.

Don't try to prove this. Convince yourself using small examples :) Especially useful are the equivalence of (a), (d) and (e). Note that a connected graph with more than $n-1$ edges contains a cycle! So if we add an edge to a tree, a cycle is created. Also, if we remove an edge from a tree it becomes disconnected. A minimum number of edges of a connected graph with n vertices is $n-1$.

Figure 11: Free trees with small number of vertices. They surely have $n-1$ edges :)

Two greedy algorithms for MST

- **Prim's algorithm** Very much like Dikstra's algorithm. Start with any vertex and grow a tree T by adding a lightest edge among the edges from T to the *outside* of T. Hence, the added edge does not make a cycle. After exactly $n-1$ iterations, we are done.

- **Kruskal's algorithm** Sort the edges in increasing order of weights. Starting from the lightest, add edges that does not make a cycle. As we already know, exactly $n-1$ edges will do.

Again, note that they are *greedy* algorithms. And they are very fast, too. (We will see implementations that runs in, basically, $O(n \log n)$ time.) Fig. 12 shows how Prim's algorithm works.