Minimum Spanning Trees

Adapted from the CLRS book slides

OVERVIEW

Problem

- A town has a set of houses and a set of roads.
- A road connects 2 and only 2 houses.
- A road connecting houses u and v has a repair cost $w(u, v)$.
- ***Goal:*** Repair enough (and no more) roads such that

  1. everyone stays connected: can reach every house from all other houses, and
  2. total repair cost is minimum.

OVERVIEW (continued)

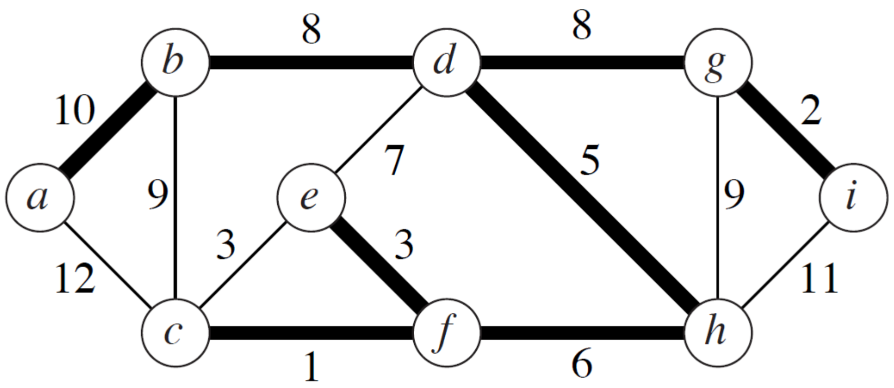
Model as a graph:

- Undirected graph $G = (V, E)$.
- ***Weight*** $w(u, v)$ on each edge $(u, v) \in E$.
- Find $T \subseteq E$ such that

1. T connects all vertices (T is a ***spanning tree***), and
2. $w(T) = \sum_{(u,v) \in T} w(u, v)$ is minimized.

OVERVIEW (continued)

A spanning tree whose weight is minimum over all spanning trees is called a ***minimum spanning tree***, or ***MST***.



In this example, there is more than one MST. Replace edge (e, f) in the MST by (c, e). Get a different spanning tree with the same weight.

OVERVIEW (continued)

Some properties of an MST:

- It has $|V| - 1$ edges.
- It has no cycles.
- It might not be unique.

BUILDING UP THE SOLUTION

- Build a set A of edges.
- Initially, A has no edges.
- As edges are added to A, maintain a loop invariant:

  Loop invariant: A is a subset of some MST.

- Add only edges that maintain the invariant. If A is a subset of some MST, an edge (u, v) is ***safe*** for A if and only if $A \cup \{(u, v)\}$ is also a subset of some MST. So add only safe edges.

GENERIC MST ALGORITHM

```
GENERIC-MST(G, w)
A = ∅
while A does not form a spanning tree
    find an edge (u, v) that is safe for A
    A = A ∪ {(u, v)}
return A
```

Use the loop invariant to show that this generic algorithm works.

Initialization: The empty set trivially satisfies the loop invariant.

Maintenance: Since only safe edges are added, A remains a subset of some MST.

Termination: The loop must terminate by the time it considers all edges. All edges added to A are in an MST, so upon termination. A is a spanning tree that is also an MST.

FINDING A SAFE EDGE

How to find safe edges?

Let's look at the example. Edge (c, f) has the lowest weight of any edge in the graph. Is it safe for $A = \emptyset$?

Intuitively: Let $S \subset V$ be any proper subset of vertices that includes c but not f (so that f is in $V - S$). In any MST, there has to be one edge (at least) that connects S with $V - S$. Why not choose the edge with minimum weight? (Which would be (c, f) in this case.)

DEFINITIONS

- A ***cut*** $(S, V - S)$ is a partition of vertices into disjoint sets V and $S - V$.
- Edge $(u, v) \in E$ ***crosses*** cut $(S, V - S)$ if one endpoint is in S and the other is in $V - S$.
- A cut ***respects*** A if and only if no edge in A crosses the cut.
- An edge is a ***light edge*** crossing a cut if and only if its weight is minimum over all edges crossing the cut. For a given cut, there can be > 1 light edge crossing it.

Theorem

Let A be a subset of some MST, $(S, V - S)$ be a cut that respects A, and (u, v) be a light edge crossing $(S, V - S)$. Then (u, v) is safe for A.

PROOF

- Let (*S*, *V*-*S*) be a cut respecting *A*, a subset of some *MST T*. All drawn edges here are part of T except for (*u*,*v*), the light edge of the cut. If (*u*,*v*) is not in the tree, replacing (*x*,*y*) with (*u*,*v*) gives a new minimum spanning tree (total weight unchanged or decreased).