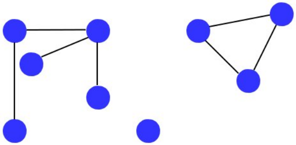
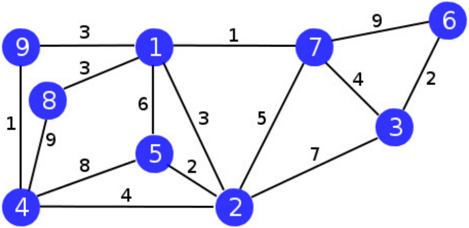
Weighted Graphs

Dr Timothy Kimber

February 2018



More Terminology

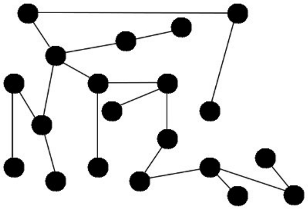


- Each edge in a weighted graph has an associated cost or weight
- We denote the weight of the edge $\{u, v\}$ by $w(u, v)$

More Terminology

Definition (Tree)

A tree is a pair (G, r) where G is a connected, acyclic graph and r is a vertex of G, called the root.

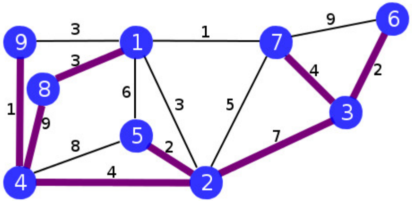


- A nonrooted tree is a connected, acyclic graph

More Terminology

Definition (Spanning Tree)

Given a graph $G = (V, E_G)$, a tree $T = (V, E_T)$ such that $E_T \subseteq E_G$ is a spanning tree for G.



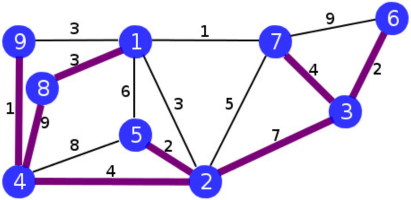
Given some network (road, phone, water ...) a minimum spanning tree (MST) is an important attribute

- Lowest cost way to connect all points

Minimum Spanning Tree

Minimium Spanning Tree Problem

Given graph $G = (V, E)$, find a new graph T such that T is an MST of G.

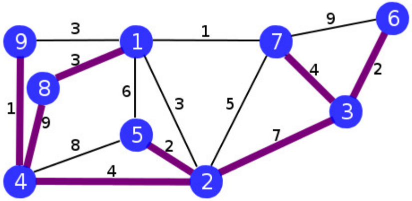


Analysis

Question

Given graph $G = (V, E)$, if you were to generate potential solutions for the MST problem, and then test them:

- What would you generate?
- What constraints apply?



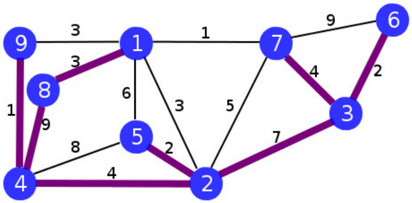
Analysis

An MST for G will comprise a set of edges $E_T \subseteq E$:

- E_T must contain $|V| - 1$ edges
- E_T must be a tree

Given this E_T will connect all vertices of G

- If such a tree also has minimal weight it is an MST



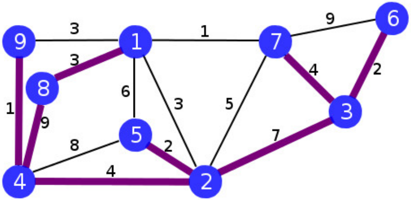
Analysis

So, finding an MST involves finding a set $E_T \subseteq E$

Exercise

Give a case-by-case definition of a function *min_edges* that returns a minimal weight set of n edges selected from array `E`.

- What are the inputs?
- What are the base cases?
- How many instances of this problem are there?



Analysis

A minimal weight set of n edges, chosen from `E[1,...,i]` is:

min_edges(Input: integer i, integer n)

$$min\_edges(i,n) = \begin{cases} \varnothing & \text{if } i = 0 \\ \varnothing & \text{if } n = 0 \\ min\_wt(\{E[i]\} \cup min\_edges(i-1, n-1), \\ \quad min\_edges(i-1, n)) & \text{otherwise} \end{cases}$$

- (Does the problem have optimal substructure?)
- $min\_edges(|E|, |V|-1)$ has $|E| \times (|V|-1)$ subproblems
- Unfortunately, *min_edges* might not produce MST. Why?

Analysis

Update to *tree_edges* and only return a set of edges that forms a tree

tree_edges(Input: integer i, set E_T)

$$tree\_edges(i, E_T) = \begin{cases} E_T & \text{if } |E_T| = |V| - 1 \\ min\_wt(tree\_edges(i-1, \{E[i]\} \cup E_T), \\ \quad tree\_edges(i-1, E_T)) & \text{otherwise} \end{cases}$$

- Subproblem now completes the tree (using reduced set of edges)
- If $\{\texttt{E[i]}\} \cup E_T$ not a tree, do not use `E[i]`
- If $|E_T| + (i-1) < |V| - 1$, insufficient edges
- $tree\_edges(|E|, \varnothing)$ still has $|E| \times (|V| - 1)$ subproblems

A New Strategy

Have seen that the problem involves this choice

- Add edge i
- Do not add edge i

Have assumed edge i could be any edge, but

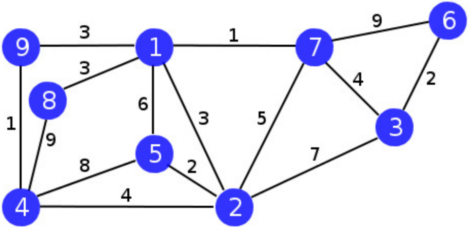
- Maybe this time a greedy approach will actually work!
- A greedy algorithm picks the 'obvious' first step
- This is called making a greedy choice
- It leaves just one subproblem to solve

So, we identify edge g, the greedy edge, and continue with $E_T \cup \{E[g]\}$

The Greedy Choice

The Greedy Choice

What greedy choices are there when computing an MST?



The Greedy Choice

- Greed is only good sometimes
- We have to show that the choice must lead to a correct solution
- (As you have seen, much easier to prove if greed is bad)

Theorem

Let G be a connected, weighted graph. If e_m is an edge of least weight in G, then e_m is in some minimum spanning tree for G.

The general method of proving that the greedy choice is OK is:

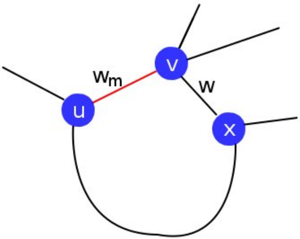
1. Suppose you have an optimal solution to the problem
2. Show that it is still optimal when the greedy choice is included

Proof

T is some MST for G

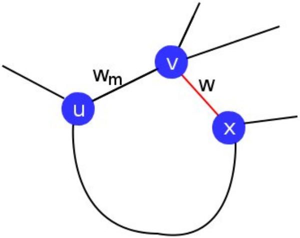
- If e_m is in T, the theorem is true
- Suppose e_m is not in T

Let $e_m = \{u, v\}$, let the path from u to v in T include the edge $\{v, x\}$, and let the weights of $\{u, v\}$ and $\{v, x\}$ be w_m and w



Proof

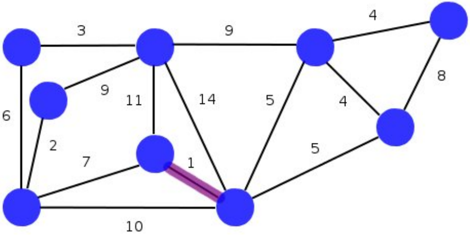
Now construct T' by removing $\{v, x\}$ from T and adding $\{u, v\}$



- Since T is a spanning tree, T' is a spanning tree
- Since T is an MST and $w_m \leq w$, T' is an MST
- e_m is in T'
- QED

Proof

Can we keep adding the next least weight edge?

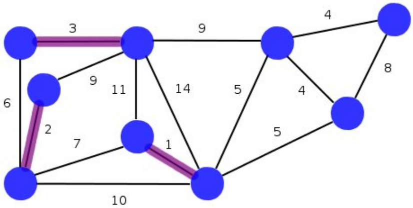


More Greed

Our greedy choice can be made more general

Theorem

Let $G = (V, E)$ be a connected, weighted graph. Let E_T be a subset of E that is part of an MST for G, and let P be a connected component in the graph (V, E_T). If E_{PQ} is the set of edges $\{u, v\}$ where exactly one of $\{u, v\}$ is in P, and e_m is an edge of least weight in E_{PQ} then e_m is in a minimum spanning tree for G.

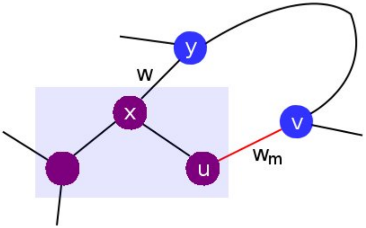


Proof

The proof is similar

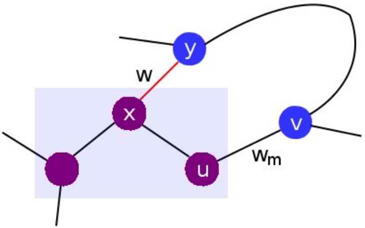
- Let T be the MST that contains E_T
- If e_m is in T, the theorem is true
- Suppose e_m is not in T

Let $e_m = \{u, v\}$, let the first edge on the path from u to v in T that is in E_{PQ} be $\{x, y\}$, and let the weights of $\{u, v\}$ and $\{x, y\}$ be w_m and w



Proof

Now construct T' by removing $\{x, y\}$ from T and adding $\{u, v\}$



- Since T is a spanning tree, T' is a spanning tree
- Since T is an MST and $w_m \leq w$, T' is an MST
- e_m is in T'
- QED