

# CPSC 320 Notes: The Futility of Laying Pipe

March 27, 2015

We're replacing UBC's aging steam heating system with a new hot water system. A set of locations needs water delivered and there's another set of intermediate points through which we can deliver water. Some of these points can be connected—at varying costs—by laying new pipe, others cannot. You'd like to figure out the cheapest way to connect every delivery location to water.

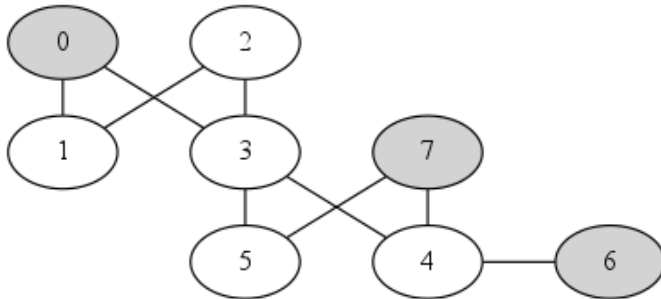
## 1 Steiner... Something-or-Others

Let's abstract and formalize this problem. We'll call it the Steiner Problem (SP).

An instance of SP is an undirected graph  $G = (V, E)$  and a subset  $S \subseteq V$  of the vertices to which we must deliver water. A solution to the instance is a subset  $E' \subseteq E$  of the edges which connects all vertices in  $S$  (and perhaps some in  $V$ ). (We could make this into an "input file format", e.g., by reading a number  $n$  indicating  $|V|$  followed by  $n$  lines, where line  $i$  lists the vertex numbers of all vertices connected to  $i$ , etc.)

Although we've ignored the costs, we could easily include them by making the edges weighted.

1. Here's a SP instance, where shaded nodes are in  $S$ . Indicate a solution to this problem.



2. Build three small, trivial example SP instances with solutions.

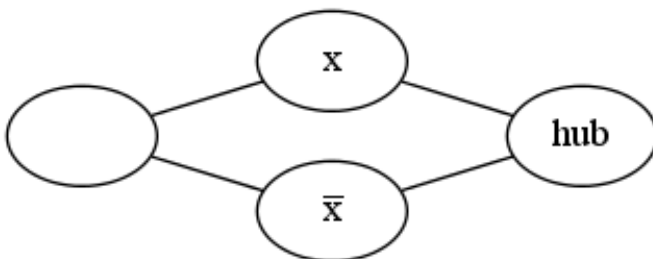
3. Build two small, non-trivial example SP instances with solutions.

4. What problem similar to this one have we solved in polynomial time? Can we just use the solution we used for that problem? If you wanted to try using the solution to that problem, how would you modify it when reporting a solution to the SP instance?
  
5. Describe a simple way to turn this into a decision problem (one where the answer is YES or NO). (Hint: think about independent set or vertex cover.)
  
6. Prove that the decision problem is in NP. (Remember: it's in NP if it's "efficiently certifiable". (Silly thought question: Is Steve "efficiently certifiable"?)

## 2 SP is NP-Complete

We've already shown that  $SP \in NP$ . To show it's NP-complete, we need to also show that it is NP-hard; that is, that it's at least as hard as every other problem in NP. We'll use 3-SAT to help us, since we already know that 3-SAT is at least as hard as every other problem in NP.

1. Which of these would show that SP is at least as hard as 3-SAT: reducing from SP to 3-SAT in polynomial time or reducing from 3-SAT to SP in polynomial time? (Hint: there is a (silly, trivial) reduction one direction or the other between checking if a list of numbers is sorted and 3-SAT in polynomial time. Whichever direction works for that is the **wrong** direction, since checking whether a list of numbers is sorted definitely only takes polynomial time.)
  
2. Here is a sketch of a "variable gadget" to help with our reduction. How can we "shade in" (put in  $S$ ) some of these vertices and choose an appropriate  $k$  (maximum number of edges in the solution) to enforce 3-SAT's choice that  $x$  or  $\bar{x}$  is true but not both?





7. Apparently unrelated question: How many edges must there be in a tree with  $n$  nodes? So... if you told someone they could use at most  $k$  edges, how many nodes could they connect?
8. Prove that your reduction is correct. Remember: your reduction is correct exactly when the answer to an instance of 3-SAT is YES if and only if the answer to the corresponding instance of SP is YES. So, your proof should proceed in two steps:
  - (a) Assume ..... Prove .....

(b) Assume ..... Prove .....

You've now shown that SP is in NP and is NP-hard (because 3-SAT—which is known to be NP-hard—is polynomial-time-reducible to SP). Therefore, SP is NP-complete!

### 3 Challenge (with one that EVERYONE should try)

1. Prove that weighted SP is NP-complete. (Everyone should complete this challenge.)
2. The “Euclidean Steiner Tree Problem” simply lists the coordinates of some points and requires connecting those points together with line segments using the shortest total length possible. Give and solve examples with three and four points, being sure to indicate why this is more like SP than like minimum spanning tree.
3. Play around with the simple-to-describe but fiendish Opaque Forest Problem: Given a convex polygon  $C$  in the (Euclidean, or other??) plane, determine the minimal set of line segments  $T$  such that any line that intersects  $C$  also intersects  $T$ . How is it related to SP? What sort of solutions to this (unsolved!) problem can you find for even very simple shapes?