CS124 Section 2

Zuzanna Skoczylas

February 2021

1   Problem 1

In a city there are N houses, each of which is in need of a water supply. It costs W_i dollars to build a well at house i, and it costs C_{ij} to build a pipe in between houses i and j. A house can receive water if either there is a well built there or there is some path of pipes to a house with a well. Give an algorithm to find the minimum amount of money needed to supply every house with water.

1.1   Solution

Create a graph with N nodes representing the houses and edges between every pair of nodes representing the potential pipes. Then add an additional "source" node which connects to house i with cost W_i so that the graph now has $N + 1$ nodes. Find an MST of this new graph.

2   Problem 2

There are n students standing in a playground trying to split into two teams to play kickball. No one cares that the teams have equal, or even nearly equal sizes, but the students do care that they are not on the same team as any of their mortal enemies. You are given a set of m enemy links, which are mutual. If person A has an enemy link with B, they absolutely cannot be on the same team. Give an algorithm to partition the students into two teams such that no enemies are on the same team. If this is not possible, return "Impossible".

2.1   Solution

To assign each student to a team, we will use a modified BFS. In the graph, each student is represented by a vertex, and each enemy link is represented by an edge between the two vertices representing the students. First, pick a student and WLOG assign him to Team 1. Then, conduct a modified BFS. At each vertex, examine all of its neighbors. If they are assigned to the same team as the vertex, return Impossible. Otherwise, assign all of the neighbors to the opposite team. Once the BFS completes, if there are still unassigned vertices,

pick another vertex and randomly assign him a team. Repeat until all vertices are assigned a team, or we return Impossible.

Correctness: If our algorithm returns a team assignment, it must be valid since our algorithm guarantees that no vertex and its neighbor will be assigned to the same team. Otherwise, when assigning teams, we would have encountered a vertex which was the same team as its neighbor and returned impossible. If our algorithm returns Impossible, there is really no solution, since two neighboring vertices must be on the same team to satisfy the other enemy links. Therefore, our algorithm is correct.

Running Time Analysis: The running time of this algorithm is $O(|E| + |V|)$ since it is just modified BFS.

3 Problem 3

(CLRS 21.3-6). Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge (one of minimum weight) crossing the cut. Conclude that this implies that if a graph has unique edge weights, then it has a unique minimum spanning tree.

3.1 Solution

Suppose we had two MST's T and T' of a graph $G = (V, E)$ that satisfied the above property. Consider any edge $t = (u, v) \in T$. Removing this edge from T separates the tree into 2 disjoint sets of vertices, S which contains u and $V S$ which contains v. S and $V - S$ represent a particular cut in the graph, so we know that any MST of G must contain the lightest edge crossing this cut. But from our assumption, we know that t is indeed the lightest edge crossing this cut, so t is in all minimum spanning trees of G. In particular, $t \in T'$. Since t was arbitrarily chosen, we know that each edge in T is also an edge in T'. Since $|T| = |T'|$,we know that $T = T'$ and the MST for G is unique. If a graph has unique edge weights, then any cut will have a unique light edge, and thus we have shown that the graph has a unique minimum spanning tree.

2