15-451/651 Algorithm Design & Analysis, Spring 2023

Recitation #6

Objectives

- Practice DFS and its applications,
- Practice reducing problems to network flows, where the max-flow/min-cut yields the optimal solution to the original problem.

Recitation Problems

1. **(DFS and Bi-Connected Components)** You have an undirected, connected graph G with n nodes labeled $\{0, \dots, n-1\}$ (representing the people living in Graph-land) and m edges (representing the contact that each person has with each other).

   Initially, we have a node n_I that we say is "infected". If a node u is infected at some time, then all its neighbors and itself i.e. $\{u\} \cup N(u)$ are also infected at the next timestep. You can imagine that at the end, all nodes that are in the same connected component as n_I get infected. You are given that n_I is not an articulation point in G.

   You, being the rising epidemiologist hired by the Graph-land city council need to remove one node (i.e. person) from the graph G other than n_I so as to minimize the total number of infections that could ever occur.

   (a) What is the algorithm to label each node as being an articulation point in graph G or not in $O(n+m)$ time?

(b) Given the articulation points for a graph, construct an $O(n+m)$ algorithm for finding the node whose removal will minimize the number of infections.

2. **(Joel Tiles)** There is an n by n grid of the letters J, O, E, and L. The goal is to form as many disjoint copies of the word JOEL as possible in the given grid. To form JOEL you start at any J, move to a neighboring O, then move to a neighboring E, and finally move to a neighboring L (you can move up, down, left, or right to get to a neighbor). The following figure shows one such problem on the left, along with three possible JOELs on the right.

J	O	J	E
L	E	O	L
E	J	O	E
L	E	O	J

J	O	J	E
L	E	O	L
E	J	O	E
L	E	O	J

J	O	J	E
L	E	O	L
E	J	O	E
L	E	O	J

J	O	J	E
L	E	O	L
E	J	O	E
L	E	O	J

Give a polynomial-time algorithm to find the solution with the largest number of JOELs. Your algorithm should consist of reducing the problem to an instance of max flow.

3. **(Cut into Teams)** CMU intramural sport season is coming up, and Maximus Flow and Minnie Cut want to form teams to compete in the intramural football game. They have a group of n students that they want to split into two teams with Maximus captaining team A and Minnie captaining team B. Each student has some preference toward being on Maximus' or Minnie's team however. In particular, Max and Min need to pay \$$A_i$ for student i to join team A and \$$B_i$ for student i to join team B. To further complicate things, some pairs of students are friends. For each friend pair, they would prefer to be on the same team. For each friend pair (i, j) that ends up on different teams, Max and Min will have to pay an extra \$$K_{i,j}$ for making them unhappy. Being college students with no money and no free time, Maximus and Minnie want to form these teams with the lowest price possible and in polynomial time. Note that the two teams do not have to be of the same size.

   (a) Reduce this to constructing a flow network where a given cut (parition of the vertices into two sets (A, B)) corresponds to assigning students to either team A or B.

(b) Now suppose each friend pair (i, j) refused to be separated and will quit if they do. How can you adjust your flow network without deleting/merging any nodes to guarantee that no such friend pair will be put on different teams?