Assignment 9

Module: Graphs

Instructions:

- Honor code: Work on this assignment with at most one partner. Between different teams, Collaboration is at level 1 [verbal collaboration only]
- Write each problem on a separate page; If a problem has multiple parts, you can write all parts on the same page, as long as you leave space in between them.

---

1. An independent set of an undirected graph $G = (V, E)$ is a subset I of V, such that no two vertices in I are adjacent (i.e., connected by an edge). A **maximal independent set** M is an independent set such that , if we were to add any additional vertex to M, it would not be independent any longer.

   (a) Draw a (connected) graph of n vertices that has a maximal IS of 1 vertex. (You will draw a small graph, but it has to be clear how to generalize to a graph of an arbitrary number of vertices.)

   (b) Draw a (connected) graph of n vertices that has a maximal IS of $n-1$ vertices. (You will draw a small graph, but it has to be clear how to generalize to a graph of an arbitrary number of vertices.)

   (c) Give an efficient algorithm that computes a maximal independent set for a graph G. Briefly justify why the set of vertices output by your algorithm is independent and maximal. Analyze the running time. [1]

2. Suppose you are given a diagram of a telephone network, which is a graph G whose vertices represent switching centers, and whose edges represent communication links between the two centers. The edges are marked by their bandwidth. The bandwidth of a path is the *minimum* bandwidth along the path. Give an algorithm that, given two switching centers a and b, will output a maximum bandwidth path between a and b.

[1] A graph may have many maximal independent sets. The largest possible IS of a graph is called the *maximum* IS. Note the difference between *maximum* and *maximal* (not any maximal set is maximum). Finding a maximal IS can be done in polynomial time (part d above), but finding the maximum IS (the largest maximal IS) is known to be hard (NPC). If your algorithm finds a *maximum* IS in polynomial time, then it's either wrong, or you'll win the Millenium Prize!