CS180 Homework 2

Due: April 29, 11:59pm

Please submit your homework on gradescope. No late submission is allowed.

1. (20 pt) Prove or disprove the following statement:

   For a graph G with n nodes (n is even), if every node of G has degree at least $n/2$, then G is connected.

2. (20 pt) A number of stories in the press about the structure of the Internet and the Web have focused on some version of the following question: How far apart are typical nodes in these networks? If you read these stories carefully, you find that many of them are confused about the difference between the *diameter* of a network and the *average distance* in a network; they often jump back and forth between these concepts as though they're the same thing.

   As in the text, we say that the distance between two nodes u and v in a graph $G = (V, E)$ is the minimum number of edges in a path joining them; we'll denote this by $dist(u, v)$. We say that the diameter of G is the maximum distance between any pair of nodes; and we'll denote this quantity by $diam(G)$.

   Let's define a related quantity, which we'll call the *average pairwise distance* in G (denoted $apd(G)$). We define $apd(G)$ to be the average, over all $\binom{n}{2}$ sets of two distinct nodes u and v, of the distance between u and v. That is,

   $$apd(G) = \left[\sum_{\{u,v\} \subseteq V} dist(u, v) \right] / \binom{n}{2}.$$

   Here's a simple example to convince yourself that there are graphs G for which $diam(G) \neq apd(G)$. Let G be a graph with three nodes u, v, w, and with the two edges $\{u, v\}$ and $\{v, w\}$. Then

   $$diam(G) = dist(u, w) = 2,$$

   while

   $$apd(G) = [dist(u, v) + dist(u, w) + dist(v, w)]/3 = 4/3.$$

   Of course, these two numbers aren't all *that* far apart in the case of this three-node graph, and so it's natural to ask whether there's always a close relation between them. Here's a claim that tries to make this precise.

   Claim: There exists a positive natural number c so that for all connected undirected graphs G (with arbitrary number of nodes), it is the case that

   $$\frac{diam(G)}{apd(G)} \leq c.$$

   Decide whether you think the claim is true of false, and give a proof of either the claim or its negation.

3. We have defined the diameter of a graph in Problem 2. Finding the diameter of a graph is not an easy problem on a general graph, but we consider a special case when G is a tree:

   (a) (20 pt) Assume G is a tree (undirected, connected graph without any cycle). Assume we are given a function $A(\cdot)$ such that given an input u, it will return a node v that maximizes dist(u, v) and the corresponding distance. There exists an algorithm that can compute the diameter of a tree using a constant calls of the function A without using any other graph traversal algorithms (e.g., BFS/DFS). Try to design such algorithm and prove the correctness of your algorithm.

   (b) (5 pt) Which graph traversal algorithm we introduced in the class can be used for implementing the function A? And what will be the overall time complexity for this diameter computing algorithm for trees in part (a)?

4. (20 pt) You're helping a group of ethnographers analyze some oral history data they've collected by interviewing members of a village to learn about the lives of people who've lived there over the past two hundred years.

   From these interviews, they've learned about a set of n people (all of them now deceased), whom we'll denote P_1, P_2, $\ldots$, P_n. They've also collected facts about when these people lived relative to one another. Each fact has one of the following two forms:

   - For some i and j, person P_i died before person P_j was born; or
   - for some i and j, the life spans of P_i and P_j overlapped at least partially.

   Naturally, they're not sure that all these facts are correct; memories are not so good, and a lot of this was passed down by word of mouth. So what they'd like you to determine is whether the data they've collected is at least internally consistent, in the sense that there could have existed a set of people for which all the facts they've learned simultaneously hold.

   Give an efficient algorithm to do this: either it should produce proposed dates of birth and death for each of the n people so that all the facts hold true, or it should report (correctly) that no such dates can exist–that is, the facts collected by the ethnographers are not internally consistent. Assume there are n people and m facts, your algorithm should run in $O(n+m)$ time.

5. (15 pt) Some of your friends have gotten into the burgeoning field of *time-series data mining*, in which one looks for patterns in sequences of events that occur over time. Purchases at stock exchanges–what's being bought are one source of data with a natural ordering in time. Given a long sequence S of such events, your friends want an efficient way to detect certain "patterns" in them–for example, they may want to know if the four events

   `buy Yahoo, buy eBay, buy Yahoo, buy Oracle`

   occur in this sequence S, in order but not necessarily consecutively.

   They begin with a collection of possible events (e.g., the possible transactions) and a sequence S of n of these events. A given event may occur multiple times in S (e.g., Yahoo stock may be bought many times in a single sequence S). We will say that a sequence S' is a subsequence of S if there is a way to delete certain of the events from S so that the remaining events, in order, are equal to the sequence S' . So, for example, the sequence of four events above is a subsequence of the sequence

   `buy Amazon, buy Yahoo, buy eBay, buy Yahoo, buy Yahoo, buy Oracle`

   Their goal is to be able to dream up short sequences and quickly detect whether they are subsequences of S. So this is the problem they pose to you: Give an algorithm that takes two sequences of events–S' of length m and S of length n, each possibly containing an event more than once–and decides in time $O(m+n)$ whether S' is a subsequence of S.

★ Homework assignments are due on the exact time indicated. Please submit your homework using the Gradescope system. Email attachments or other electronic delivery methods are not acceptable. To learn how to use Gradescope, you can:

  – 1. Watch the one-minute video with complete instructions from here: https://www.youtube.com/watch?v=-wemznvGPfg
  – 2. Follow the instructions to generate a PDF scan of the assignments: http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf
  – 3. **Make sure you start each problem on a new page.**

★ We recommend to use LaTeX, LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receiver fewer points.

★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.