Write your solution using \LaTeX. Submit this homework using handin41. This is a partnered homework. There are no assigned partners this week, but partners are still required. It is up to you to choose which partner to work with. If you are having trouble finding a partner, please first post on Piazza, then email me if you have trouble finding a partner.

It is still OK to discuss approaches with others at a high level, but most of your discussions should be just with your lab partner. The only exception to this rule is work you’ve done with someone while in lab. In this case, note who you’ve worked with and what was solved during lab.

If there are questions about academic integrity, please visit the section on Academic Integrity on the course website (www.cs.swarthmore.edu/~brody/cs41/s15/expectations.php). If you still have questions, please contact me.

Note: Make sure your homework includes the names of both lab partners. Only one partner should submit files. Make sure the files you submit are in the cs41/hw/xx directory before calling handin41.

Note: The algorithms in problems (1) and (2) should run in polynomial time. Provide pseudocode for the algorithms in problem (1); an English description for problem (2) suffices. For either problem, you do not need to give full formal proofs of correctness or runtime analysis, but you should give enough justification to convince the graders that your algorithms are correct and reasonably efficient.

1. The following algorithm detects cycles in an undirected graph, returning TRUE if and only if a cycle exists.

\begin{verbatim}
DETECTCycles(G)
1 Mark each node as not visited
2 for i = 1...n
3     if i not visited
4         if CycleDFS(G, i, NULL)
5             return TRUE
6 Return FALSE

CycleDFS(G, v, parent)
1 Mark v as visited.
2 for each neighbor u of v
3     if u ≠ parent
4         if u already visited
5             return TRUE
6     else if CycleDFS(G, u, v)
7         return TRUE
8 return FALSE
\end{verbatim}

This algorithm does not correctly detect cycles in directed graphs.
(a) Give a specific directed graph for which this algorithm does not work.

(b) Modify this algorithm (or create a new algorithm) which detects cycles in directed graphs. The input for your algorithm is a directed graph $G = (V, E)$. Your algorithm should return \textsc{true} if and only if $G$ has a cycle.

(c) Modify your algorithm so it returns a cycle if the graph contains one.

2. (Kleinberg and Tardos, 3.12) 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 have lived there over the past two hundred years.

From these interviews, they’ve learned about a set of $n$ people (all 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 very 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 \textit{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.

3. ICPC Problem 6484 (Ping!).