CENG 567 Design and Analysis of Algorithms Fall 2020

Homework 2 - Graphs

Released: 6/11/2020 Due: 15/11/2020

Instructions. You may work with other students, but you must individually write your solutions **in your own words**. If you work with other students or consult outside sources (such as Internet/book), cite your sources.

If you are asked to design an algorithm, provide: (a) the pseudocode or precise description in words of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submissions. Submit a pdf file through odtuclass. LaTeX or Word typed submission is required.

1. **Checking consistency of judgments. (shorter rewording of K&T Ch 3 Ex 4)**
Given a collection of n butterflies, amateur lepidopterists want to divide them into two groups: those that belong to specie A and those that belong to specie B. They study each pair of specimens carefully side by side. They label each pair of specimens i and j either "same" or "different". They may also call the pair as "ambiguous" if they have no judgment on a given pair. Given n specimens and m judgments (either "same" or "different") they want to know if these judgments are consistent with the idea that each butterfly is from one of the species A or B. In other words the m judgments will be consistent if it is possible to label each specimen either A or B in such a way that for each pair (i, j) labelled "same", it is the case that i and j have the same label; and for each pair (i, j) labelled "different", it is the case that i and j have the different labels.

Give an algorithm with running time $O(m + n)$ that determines whether the m judgments are consistent.

2. **Reachability** (Cormen, Leiserson, Rivest, Stein, problem 22-4)
Let $G = (V, E)$ be a directed graph in which each vertex $u \in V$ is labeled with a unique integer $L(u)$ from the set $\{1, 2, \ldots, |V|\}$. For each vertex $u \in V$, let $R(u) = \{v \in V : u \rightsquigarrow v\}$ be the set of vertices that are reachable from u. Define $\min(u)$ to be the vertex in $R(u)$ whose label is minimal, i.e., $\min(u)$ is the vertex v such that $L(v) = \min\{L(w) : w \in R(u)\}$. Give an $O(V + E)$ time algorithm that computes $\min(u)$ for all vertices $u \in V$.