CENG 567 Design and Analysis of Algorithms Fall 2022

Homework 2

Released: 26/10/2022 Due: 6/11/2022

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) a description of the algorithm in English and, if helpful pseudocode,
(b) at least one worked example or diagram to show more precisely how your algorithm works,
(c) a proof (or indication) of the correctness of the algorithm,
(d) an analysis of the implementation of the algorithm.

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

1. Visiting all edges. Suppose you have a network of two-way streets.
(a) Show that you can drive along these streets so that you visit all streets and you drive along each side of every street exactly once.
(b) Suppose you drive in such a way that at each intersection, you do not leave by the street you used to enter that intersection unless you have previously left via all other streets from that intersection. Does this give a valid way to solve part (a)? If yes, prove this, and if not, construct a counterexample.

2. Connectivity.
(a) Suppose you are given a directed graph G. We create a new graph H as follows: First, we find all the strongly connected components of G. For each strongly connected component C_i in G, we create a node v_i in H. Second, if there is an edge from C_i to C_j in G, we create an edge from v_i to v_j in H. Prove that H is a DAG.

(b) If G is a forest (acyclic graph) with n nodes, k edges, and c components, prove that $c = n - k$.

3. Job scheduling. There is a set of n employees $E_1, ..., E_n$, and a set of tasks $T_1, ..., T_n$. Each employee must be assigned to one task, and every task must be assigned to an employee. Each task T_i has a difficulty level d_i, and each employee has a skill level s_i. The goal is to match employees to tasks such that the average difference between each employee's skill level and his or her task's difficulty level is minimized. In other words, if employee E_i is matched to task $T_{f(i)}$, we wish to minimize

$$\frac{1}{n} \sum_{i=1}^{n} |s_i - d_{f(i)}|$$

(a) Consider the algorithm that begins with with employee E_1, and assigns that employee the task with the difficulty level that is closest to E_1's skill level. Remove E_1 and the selected task from the pool, and assign employee E_2 the task with difficulty level closest to E_2's skill level. Repeat until all employees and tasks are matched. Give a counterexample showing when this method fails to find the optimal solution.

(b) Describe an $O(n \log n)$ algorithm to find the optimal solution.

4. Spanning Trees and DFS. Consider a connected undirected graph $G = (V, E)$, a spanning tree T of G and a vertex v. Design an $O(|V| + |E|)$ algorithm to determine whether T is a valid DFS tree of G rooted at v, that is, determine whether T can be the output of DFS starting from v, under some order of the edges.