

Homework 2

Released 2/10/2020

Due 2/24/2020 11:59pm in Gradescope

Instructions. You may work in groups, but you must write solutions yourself. List collaborators on your submission. Also list any sources of help (including online sources) other than the textbook and course staff.

If you are asked to design an algorithm, please 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. Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope.

1. **Bipartite Matchings (10 points)** Let G be a bipartite graph, with two sets of vertices A and B , each of size n , and some edges, each with one endpoint in A and one in B . A perfect matching in this graph is a subset of the edges such that every vertex in A and B is an endpoint of exactly one edge.

A student recalls that the Gale-Shapley algorithm finds perfect matchings, and proposes the following. Give each vertex a preference list on the other set of vertices that puts all of its neighbors ahead of its non-neighbors. Use Gale-Shapley to find a stable matching M for this list. If all matched pairs in M correspond to graph edges, this is a perfect matching in G . Otherwise, the student claims, G has no perfect matching, because if there were one, but M matches some nodes to non-neighbors, then M would be unstable and not the result of Gale-Shapley. Is the student correct? Argue why or why not.

2. **Greedy Partition (10 points)** Given n positive integers, we want to partition them into two sets A and B so that the sums S_A and S_B of their elements are as evenly balanced as possible, i.e., there is no other split (A', B') such that $|S_{A'} - S_{B'}| < |S_A - S_B|$. We use the following greedy algorithm: sort the values in decreasing order, start with empty A and B and always put the next value into the set with the lowest sum (choose any set if their sums are equal).

Does this algorithm always produce the most balanced split? Prove it, or find a counterexample.

3. **Checking Tasks (20 points)** Consider n tasks, each with starting and finishing times (real numbers). We'd like to check that each task is running properly. At a timepoint t we can simultaneously check all tasks that are running at that time (for which t is strictly between task start and finishing time).

Give an algorithm that finds the smallest set of timepoints so that each task is checked at least once.

4. **Surfboard Rental (20 points)** A group of n tourists with heights h_i go to a rental place with n surfboards, each of length l_j . Give an algorithm that will assign each tourist i their own surfboard $a(i)$ such that the total size difference between tourists and their boards $\sum_{i=1}^n |h_i - l_{a(i)}|$ is minimized.

5. **Board Coloring (20 points)** We want to color an $n \times n$ board black and white. Arrays $R[1..n]$ and $C[1..n]$ tell us the number (between 0 and n) of squares to be colored black in each row and column, respectively.

Give an algorithm that either colors the board as specified, or correctly answers that this is impossible.

6. **Transforming Trees (20 points).** Given a connected graph G , we call spanning trees T and T' *one step apart* if we get T' by removing an edge from T and adding an edge of T' . Design an algorithm that finds the smallest number of steps needed to get from a given spanning tree T_1 to another one T_2 .

7. **(0 points).** How long did it take you to complete this assignment?