University of Washington February 2, 2024
Department of Computer Science and Engineering
CSE 421, Winter 2024

Homework 5, Due Wednesday, February 7, 11:59 pm, 2024

Turnin instructions: Electronic submission on gradescope using the CSE 421 gradescope site. Submit the assignment as a PDF, with separate pages for different numbered problems. Problems consisting of multiple parts (e.g., 2a, 2b) can be submitted on the same page.
For all of these problem, provide a justification as to why your algorithm correctly solves the problem. Give and justify the run time for your algorithms.

Problem 1 (10 points):

Suppose A is an array of n integers that is a strictly decreasing sequence, followed by a strictly increase sequence such as $[12, 9, 8, 6, 3, 4, 7, 9, 11]$. Give an $O(\log n)$ algorithm to find the minimum element of the array. Justify your algorithm is correct.

Problem 2 (10 points):

Let A and B be two sorted arrays of integers, each of length n. Show how you can find the median of the combined set of elements in $O(\log n)$ comparisons. (As in the Median algorithm discussed in lecture, you will need to solve the Select the k-th largest problem.) Justify your algorithm is correct.

Problem 3 (10 points) Weighted Independent Set on a Path:

The weighted independent set problem is: Given an undirected graph $G = (V, E)$ with weights on the vertices, find an independent set of maximum weight. A set of vertices I is independent if there are no edges between vertices in I. This problem is known to be NP-Complete.

For a simpler problem, consider a graph that is a path, where the vertices are $v_1, v_2, \ldots, v_n$, with edges between v_i and v_{i+1}. Suppose that each node v_i has an associated weight w_i. Give an algorithm that takes an n vertex path with weights and returns an independent set of maximum total weight. The run time of the algorithm should be polynomial in n. Justify your algorithm is correct.

Problem 4 (10 points) Task Choice :

Suppose that each week you have the choice of a high stress task, a low stress task, or no task. If you take a high stress task in week i, you are not allowed to take any task in week $i+1$. For n weeks, the high stress tasks have payoff $h_1, \ldots, h_n$, and the low stress tasks have payoff $l_1, \ldots, l_n$, and not doing a task has payoff 0. Give an algorithm which given the two lists of payoffs, maximizes the value of tasks that are performed over n weeks. The run time of the algorithm should be polynomial in n. Justify your algorithm is correct.

Problem 5 (10 points) Word segmentation:

(This problem is based on problem 5 on Page 316 of the text without the excessive verbiage.) The word segmentation problem is: given a string of characters $Y = y_1 y_2 \dots y_n$, optimally divide the string into consecutive characters that form words. (The motivation is that you are given a text string without spaces and have to figure out what the words are. For example, *"meetateight"* could be *"meet ate ight", "me et at eight"* or *"meet at eight".*) The problem is to find the best possible segmentation. We assume we have a function *Quality* which returns an integer value of the goodness of a word, with strings that correspond to words getting a high score and strings that do not correspond to words getting a low score. The overall quality of a segmentation is the sum of the qualities of the individual words.

Give a dynamic programming algorithm to compute the optimal segmentation of a string. You can assume that calls to the function *Quality* take constant time and return an integer value. What is the runtime of your algorithm? Justify your algorithm is correct.