

Assignment 3

Due: Monday, 16 November 2020, 3:30 p.m. (before the start of the lecture)

To submit the assignment, upload a .pdf-, .doc-, .docx-, .txt-, or .jpg-file with your solutions to Blackboard before the deadline. The filename should start with “A3_99999999Z” (replace “99999999Z” by your student ID number). Include your name and student ID number at the top of each page. Scanned, handwritten solutions are OK as long as they are easily readable. You may specify your algorithms in pseudocode or in plain English.

The maximum possible score is 15 marks. Late submissions will not be accepted. The assignment is to be completed individually, i.e., every student must turn in his or her own solutions. Students are allowed to discuss solution strategies with other students but they are not allowed to copy each others' answers.

1. Suppose randomized QUICKSORT is applied to an array $A[1..n]$ of distinct numbers, where $n \geq 4$.
 - a) What is the probability that the algorithm will compare the smallest number in A and the largest number in A to each other at least once during its execution? Justify your answer. (1 mark)
 - b) What is the probability that the algorithm will compare the second smallest number in A and the third smallest number in A to each other at least once during its execution? Justify your answer. (1 mark)
2. In this question, we shall use the following definitions. Let $G = (V, E)$ be a graph and let k be a positive integer. A *relaxed k -coloring of G* is any partition of V into disjoint subsets V_1, V_2, \dots, V_k . In a relaxed k -coloring of G , if x belongs to V_i then x is said to have the color i . Note that in contrast to the other graph colorings we have seen earlier in the course, a relaxed k -coloring of G allows two vertices that are connected by an edge to have the same color.

An edge $\{x, y\}$ in E is called a *good edge* if x and y have different colors. Consider the following maximization problem:

Input: An undirected graph $G = (V, E)$.

Output: A relaxed 4-coloring of G with as many good edges as possible.

Prove that for any instance of this problem, if each vertex in V is assigned to one of V_1, V_2, V_3, V_4 chosen uniformly at random and independently of all other vertices, then the expected number of good edges is at least $\frac{3}{4}$ times the number of good edges in an optimal solution. (2 marks)

3. Do a literature search to find a published conference paper or journal article that uses the *potential method* to analyze the time complexity of an algorithm, and summarize the result. In your answer to this question, write the following:
 - Bibliographic information (the names of the authors, the title of the paper/article, the conference or journal name, the volume and page numbers, and the year of publication).
 - A short description of the problem considered and the result of the analysis.
 - The definition of the potential function used.

The analysis that you choose should not be described in the textbook or the tutorials. You do not need to include any calculations or technical details of the algorithm in your answer. (3 marks)

(Continued \longrightarrow)

4. Two strings S_1 and S_2 are *anagrams* if S_2 can be obtained by rearranging the characters of S_1 . For example, LEMON and MELON are anagrams. Design a fast algorithm for the following problem and analyze its time complexity.

Input: A string T and a string P , both over the alphabet $\{A, B, C, \dots, Z\}$.

Output: The set of shifts in T at which P or an anagram of P occurs, i.e., every integer $s \in \{0, 1, \dots, |T| - |P|\}$ such that $T[(s+1)..(s+|P|)] = P$ or $T[(s+1)..(s+|P|)]$ is an anagram of P .

For example, if the input is $T = \text{LIMESANDLEMONS}$ and $P = \text{MELON}$ then the output is $\{8\}$ because $T[9..13]$ is an anagram of P . To get full marks, the worst-case running time of your solution must be $O(|T| + |P|)$. (3 marks)

5. Describe an efficient algorithm for the following problem.

Input: Two strings T and P , where T contains no “?”-symbols and P contains exactly one “?”-symbol.

Output: “YES” if there exists an integer s such that for each $i \in \{1, 2, \dots, |P|\}$, either $P[i] = T[s+i]$ or $P[i] = \text{“?”}$ holds; “NO” otherwise.

For example, if $T = \text{STRENGTH}$ and $P = \text{R?NG}$ then the output should be “YES”, but if $T = \text{STRENGTH}$ and $P = \text{R?G}$ then the output should be “NO”. You may use the linear-time algorithm for the Exact String Matching Problem from the lectures as a procedure (you do not need to give its pseudocode here).

Also, analyze the time complexity of your algorithm. To get full marks, its worst-case running time must be $O(|T| + |P|)$. (3 marks)

6. The efficient version of the rooted-trees representation of disjoint sets from Lecture 9 incorporates a path compression heuristic in the procedure FIND-SET. The procedure is defined as:

FIND-SET(x)

if $x \neq x.p$

$x.p = \text{FIND-SET}(x.p)$

return $x.p$

Rewrite the pseudocode so that the procedure does not use recursion. After running your modified procedure, the $x.p$ -values should have the same values as if the recursive FIND-SET above had been used. (2 marks)