Exam in Algorithms & Data Structures 3 (1DL481)

Prepared by Pierre Flener

Friday 14 March 2025 from 14:00 to 17:00 at Bergsbrunnagatan 15

Materials: This is a ***closed***-book exam, drawing from the book *Introduction to Algorithms* by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, published in 4th edition by The MIT Press in 2022, and denoted by CLRS4 here. ***No*** means of help are allowed.

Instructions: Question 1 is ***mandatory***: you must earn ***at least half*** of its points in order to pass this exam (see **Grading** below). Start each answer on a new sheet, and indicate there the question number. Your answers must be written in English. Unreadable, unintelligible, and irrelevant answers will not be considered. Provide only the requested information and nothing else, but always show ***all*** the details of your reasoning, unless explicitly not requested, and make explicit ***all*** your additional assumptions. Do ***not*** write anything into the following table:

Question	Max Points	Your Mark
1	6	
2	4	
3	4	
4	6	
Total	20	

Help: No teacher will attend the exam. If you think a question is unclear, then explain your difficulty with the question and state the assumption that underlies your answer.

Grading: Your grade is as follows when your exam mark is e points, including ***at least*** 3 points on Question 1:

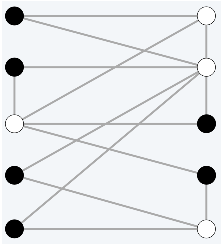
Grade	Condition
5	$18 \leq e \leq 20$
4	$14 \leq e \leq 17$
3	$10 \leq e \leq 13$
U	$00 \leq e \leq 09$

Identity: Your anonymous exam code:

1	D	0	7	E	-	0	0			-			

Question 1: NP-Completeness (*mandatory question*!) (6 points)

Problem 34-1a: An ***independent set*** of a graph $G_2 = (V_2, E_2)$ is a subset $V_2' \subseteq V_2$ of vertices such that each edge in E_2 is incident on at most one vertex in V_2'. The ***independent-set problem*** is to find a maximum-size independent set in G_2.



For example, to the left is an independent-set instance (V_2, E_2), where the vertices of V_2 are the 10 black and white points and the edges of E_2 are depicted by the lines. A maximum-size independent set consists of the 6 black vertices: each edge in E_2 is incident on at most one black vertex.

Formulate a related decision problem for the independent-set problem, and prove that it is NP-complete, by using a *single* reduction, *directly* from the ***clique problem***, whose decision version asks if an undirected graph $G_1 = (V_1, E_1)$ contains a clique of size k_1, that is a subset $V_1' \subseteq V_1$ of k_1 vertices, each pair of which being connected by an edge in E_1.

You ***must earn at least half of the points*** of this question.
You ***must*** use the identifiers G_1, V_1, E_1, V_1', k_1, G_2, V_2, E_2, and V_2', with their meanings above.

Question 2: Probabilistic Analysis (4 points)

Exercise 5.2-6: Let $A[1:n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the index pair (i, j) is called an ***inversion*** of A. For example, the inversions of the array $\langle 2, 3, 8, 6, 1 \rangle$ are $(1, 5)$, $(2, 5)$, $(3, 4)$, $(3, 5)$, and $(4, 5)$. Suppose that the elements of A form a uniform random permutation of $\langle 1, 2, \ldots, n \rangle$. Use indicator random variables to compute the expected number of inversions.

Question 3: Amortised Analysis (4 points)

Exercise 16.3-2: Use a *potential method of analysis* (also known as the *physicist's method*) to determine the amortised cost per operation for a sequence of n operations on a data structure in which the ith operation costs i if i is an exact power of 2, and 1 otherwise.

Question 4: Approximation Algorithms (6 points)

Exercise 35.4-2: The input to the ***MAX-3-CNF satisfiability problem*** is the same as for 3-CNF satisfiability (which asks whether a conjunction of clauses, each of exactly three distinct literals, is satisfiable; a *literal* is an occurrence of a Boolean variable or its negation), and the goal is to return an assignment of the variables that maximises the number of satisfied clauses. The ***MAX-CNF satisfiability problem*** is like the MAX-3-CNF satisfiability problem, except that it does not restrict each of the m clauses C_i to have exactly three literals. Give a randomised 2-approximation algorithm for the MAX-CNF satisfiability problem. Denote its produced number of satisfied clauses by M and the actual maximum number of satisfiable clauses by M^*. A high-level argument for polynomial time suffices.

You ***must*** use the identifiers m, C_i, M, and M^* with their meanings above.
You ***must*** first clarify whether you use the CLRS4 convention ($M \leq M^* \leq 2 \cdot M$) or the Princeton convention ($0.5 \cdot M^* \leq M \leq M^*$) for this maximisation problem.