Exam in Algorithms & Data Structures 3 (1DL481)

Prepared by Pierre Flener

Friday 17 March 2023 from 14:00 to 17:00 in hall 2 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	8	
2	3	
3	6	
4	3	
Total	20	

Help: Unfortunately, no teacher can attend the exam.

Grading: Your grade is as follows when your exam mark is e points, including ***at least*** 4 points on Question 1:

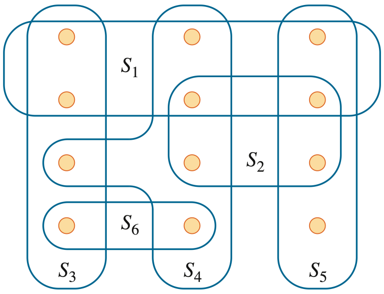
Grade	Condition
5	$18 \leq e \leq 20$
4	$14 \leq e \leq 17$
3	$10 \leq e \leq 13$
U	$00 \leq e \leq 09$

Identity: Your anonymous exam code:

1	3	0	5	C	-	0	0			-			

Question 1: NP-Completeness (*mandatory question*!) (8 points)

Exercise 35.3-2: An instance $(X, \mathcal{F})$ of the ***set-covering problem***, which is an optimisation problem, consists of a finite set X and a family $\mathcal{F}$ of subsets S of X, such that every element of X belongs to at least one subset in $\mathcal{F}$, that is $X = \bigcup_{S \in \mathcal{F}} S$. The problem is to find a minimum-size subfamily $\mathcal{C} \subseteq \mathcal{F}$ whose members cover all of X, that is $X = \bigcup_{S \in \mathcal{C}} S$. We say that any such subset $\mathcal{C}$ is a ***cover*** of X.



To the left is a set-covering instance $(X, \mathcal{F})$, where X consists of the 12 grey points and $\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$. Each set $S_i \in \mathcal{F}$ is outlined by a contour. A minimum-size cover of X is $\mathcal{C} = \{S_3, S_4, S_5\}$, with size 3.

Show that the decision version of the set-covering problem is NP-complete by using a *single* reduction, *directly* from the decision version of the ***vertex-cover problem***, which asks to find a vertex cover of minimum size in a given undirected graph $G = (V, E)$, that is a minimum-size subset $V' \subseteq V$ such that if $(u, v) \in E$, then either $u \in V'$ or $v \in V'$ (or both).

You must earn ***at least half*** of the points of this question. Start your answer on a new sheet. You ***must*** use the identifiers X, $\mathcal{F}$, S, S_i, $\mathcal{C}$, G, V, E, and V' of the question.

Question 2: Randomised Algorithms (3 points)

Exercise 5.3-4: Professor Knievel suggests the following procedure to generate a uniform random permutation of its input array A of length n:

```
PERMUTE(A, n)
let B[1 : n] be a new array
offset = RANDOM(1, n)        // RANDOM(ℓ, u) returns a random number from ℓ to u inclusive
for i = 1 to n
    dest = i + offset
    if dest > n
        dest = dest − n
    B[dest] = A[i]
return B
```

Give the computation of the probability of each element $A[i]$ of winding up in any particular position in B. Explain whether the resulting permutation is uniformly random or not.

Start your answer on a new sheet.

Question 3: Approximation Algorithms (6 points)

Problem 35-3: ***Weighted set-covering problem***: Suppose that sets have weights in the set-covering problem (see Question 1), so that each set S_i in the family $\mathcal{F}$ has an associated weight w_i. The ***weight*** of a cover $\mathcal{C}$ is $\sum_{S_i \in \mathcal{C}} w_i$. The goal is to determine a minimum-weight cover. CLRS4 handles the case in which $w_i = 1$ for all i by the following greedy approximation algorithm:

GREEDY-SET-COVER$(X, \mathcal{F})$

$U_0 = X$ // each set U_i has the remaining uncovered elements of X
$\mathcal{C} = \emptyset$ // the set $\mathcal{C}$ is the cover being constructed
$i = 0$
while $U_i \neq \emptyset$
    select an $S \in \mathcal{F}$ that maximises $|S \cap U_i|$
    $U_{i+1} = U_i - S$
    $\mathcal{C} = \mathcal{C} \cup \{S\}$
    $i = i + 1$
return $\mathcal{C}$

On the instance of Question 1, this algorithm produces a sub-optimal cover of size 4 by selecting either the sets S_1, S_4, S_5, and S_3 or the sets S_1, S_4, S_5, and S_6, in order.

Show how to generalise this algorithm in a natural manner to provide an approximate solution for any instance of the weighted set-covering problem. Does the CLRS4 analysis for the unweighted case generalise, establishing that your algorithm also is a polynomial-time $\mathcal{O}(\lg |X|)$-approximation algorithm? A yes/no answer with a high-level argument suffices for this last task.

Start your answer on a new sheet. You ***must*** use the identifiers X, $\mathcal{F}$, S, S_i, w_i, $\mathcal{C}$, and U_i of the question.

Question 4: Amortised Analysis (3 points)

Exercise 16.2-1: You perform a sequence of PUSH and POP operations on a stack whose size somehow never exceeds k. After every k operations, a COPY operation of the entire stack is made automatically, for backup purposes, but without emptying the stack. Use an *accounting method of amortised analysis* in order to show that the cost of n stack operations, including copying the stack, is $\mathcal{O}(n)$ by assigning suitable amortised costs to the various stack operations.

Start your answer on a new sheet.