National University of Singapore

School of Computing

CS3230 - Design and Analysis of Algorithms

Midterm Test

(Semester 1 AY2024/25)

Time Allowed: 100 minutes

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. This assessment paper contains TWO (2) sections.
   It comprises TWELVE (12) printed pages, including this page.
3. This is an **Open Book** Assessment.
4. For Section A, use the boxes at page 9 (use 2B pencil).
   You will still need to hand over the entire paper as the MCQ section will not be archived.
   For Section B, answer **ALL** questions within the **boxed space**.
   If you leave the boxed space blank, you will get automatic free ~~10% of the mark allocated~~.
   Unfortunately for us and fortunately for students, SoftMark smallest score is 0.5 marks.
   However, if you write at least a single character and it is totally wrong, you will get 0 mark.
   You can use either pen or pencil. Just make sure that you write **legibly**!
5. Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
   Read all the questions first! Some questions might be easier than they appear.
6. You can assume that all **logarithms are in base** 2.
7. The total marks of this paper is 40 marks.
   It will then be scaled to 20% of the course weightage.

A MCQs ($15 \times 1 = 15$ marks)

1. Which of the following statements is **TRUE**?

   a). $O(n^{1/\log\log n}) \subseteq \omega(\sqrt{n})$

   b). $O(n^{1/\log\log n}) \subseteq \Theta(\sqrt{n})$

   c). $\Omega(n^{1/\log\log n}) \subseteq \omega(\sqrt{n})$

   d). $\Omega(n^{1/\log\log n}) \subseteq \Theta(\sqrt{n})$

   e). None of the above

2. Which of the following statements is **TRUE**?

   a). If $f(n) \in O(g(n))$, then $2^{f(n)} \in O(2^{g(n)})$

   b). If $f(n) \in O(g(n))$, then $2^{g(n)} \in O(2^{f(n)})$

   c). If $f(n) \in O(g(n))$, then $g(n)^{f(n)} \in O(f(n)^{g(n)})$

   d). If $f(n) \in O(g(n))$, then $f(n)^{g(n)} \in O(g(n)^{f(n)})$

   e). None of the above

3. Consider the recurrence relation: $T(n) = 1.999 \cdot T(n/2) + n$.
   Which of the following statements is **TRUE**?

   a) $T(n) \in o(n)$

   b) $T(n) \in \Theta(n)$

   c) $T(n) \in \Theta(n \log n)$

   d) $T(n) \in \omega(n \log n)$

   e) None of the above

4. Consider the recurrence relation: $T(n) = a \cdot T(n/b) + n^d$ where $b > 1$ and $d \geq 0$.
   Which of the following statements is **TRUE**?

   a). If $a = 1$, we will always fall into case 3 of master theorem.

   b). If $a = 1$, we will never fall into case 1 of master theorem.

   c). If $a > b$, we will always fall into case 1 of master theorem.

   d). If $a > b$, we will never fall into case 3 of master theorem.

   e). None of the above

5. Which of the algorithms that have been discussed in class is **not** a Divide and Conquer algorithm?
   a). Strassen's
   b). Karatsuba's
   c). Insertion Sort
   d). Stooge Sort
   e). None of the above

6. Let $gcd(a, b)$ denote the greatest common divisor between two non-negative integers a and b. Consider the following recursive algorithm to compute $gcd(a, b)$.
   - If $b = 0$, the algorithm returns a (indeed, in this case, $gcd(a, b) = a$).
   - Otherwise, the algorithm returns $gcd(b, a\%b)$, which is computed recursively (indeed, in this case, $gcd(a, b) = gcd(b, a\%b)$).

   Which of the following statements is **FALSE**?
   a). For all a and b with $n = a + b$, the recursive algorithm to compute $gcd(a, b)$ runs in $o(\log n)$ in the worst case.
   b). $gcd(553, 511) = 7$
   c). $gcd(10000, 17) = 1$
   d). $gcd(0, b) = b$
   e). There exist a and b with $n = a + b$ such that the recursive algorithm to compute $gcd(a, b)$ runs in $\Omega(\log n)$ time in the worst case.

7. MCQ 7 has a bit of technical issue and is thus not archived.

8. Consider a variant of the deterministic quicksort algorithm where the pivot selection is done using a function f that maps each positive integer k to an index in $\{1, 2, \ldots, k\}$: If the size of the current array A is k, the algorithm selects $f(k)$-th element $A[f(k)]$ as the pivot. Let $T_f(n)$ be the worst-case time complexity of the deterministic quicksort algorithm on an array of n distinct numbers already in the sorted order where the pivot selection is done using f. Let $f_1(k) = 1$, $f_2(k) = \lceil k/2 \rceil$, and $f_3(k) = k$. Which of the following statements is **TRUE**?

   a). $T_{f_1}(n) = \Theta(n)$, $T_{f_2}(n) = \Theta(n)$, and $T_{f_3}(n) = \Theta(n)$

   b). $T_{f_1}(n) = \Theta(n)$, $T_{f_2}(n) = \Theta(n \log n)$, and $T_{f_3}(n) = \Theta(n^2)$

   c). $T_{f_1}(n) = \Theta(n^2)$, $T_{f_2}(n) = \Theta(n \log n)$, and $T_{f_3}(n) = \Theta(n)$

   d). $T_{f_1}(n) = \Theta(n^2)$, $T_{f_2}(n) = \Theta(n^2)$, and $T_{f_3}(n) = \Theta(n^2)$

   e). None of the above

9. Consider an execution of the randomized quicksort algorithm on an array of n distinct numbers. Define $\mathcal{E}_{i,j}$ as the event that the algorithm compares the ith smallest element with the jth smallest element in the array during the algorithm. For the case of $1 \le i < k < l < j \le n$, what is the value of $\mathbf{Pr}[\mathcal{E}_{i,j}$ and $\mathcal{E}_{k,l}]$?

   a). $\frac{4}{(j-i+1)(l-k+1)}$

   b). $\frac{2}{(j-i+1)(l-k+1)}$

   c). $\frac{2}{j-i+1} + \frac{2}{l-k+1}$

   d). $\frac{2}{j-i+1} + \frac{2}{l-k+1} - \frac{4}{(j-i+1)(l-k+1)}$

   e). None of the above

10. Place 100 balls into 100 bins. Each ball placement is done independently and uniformly at random. Let X_i be the number of balls in the i-th bin. Let $\mathcal{E}_{i,j}$ be the event that the i-th ball is in the j-th bin. What is the value of $\mathbf{E}[X_1 | \mathcal{E}_{1,1}] + \mathbf{E}[X_1 | \mathcal{E}_{1,1} \wedge \mathcal{E}_{2,1}]$?

    a). 5

    b). 4.99

    c). 4.98

    d). 4.97

    e). None of the above.

11. Let $G = (V, E)$ be a complete graph over 11 vertices. Each vertex selects one of its 10 incident edges independently and uniformly at random. Let $E^* \subseteq E$ be the set of all edges selected by at least one of the two endpoints. What is the expected value of $|E^*|$?

    a). 11

    b). 10.55

    c). 10.45

    d). 10

    e). None of the above.

12. A palindrome is a string that reads the same forward and backward.
    We define $\mathbf{lps}(S)$ as the *length* of the longest palindrome subsequence of $S = s_1 s_2 \cdots s_{n-1} s_n$.
    Which of the following statements is **TRUE**?

    a). If $s_1 = s_n$, then $\mathbf{lps}(s_2 \cdots s_n) = \mathbf{lps}(s_1 \cdots s_{n-1})$.

    b). If $s_1 \neq s_n$, then $\mathbf{lps}(s_2 \cdots s_n) = \mathbf{lps}(s_1 \cdots s_{n-1})$.

    c). If $s_1 \neq s_n$, then $\mathbf{lps}(s_2 \cdots s_{n-1}) = \mathbf{lps}(s_1 \cdots s_n)$.

    d). If $\mathbf{lps}(s_2 \cdots s_{n-1}) = \mathbf{lps}(s_1 \cdots s_n)$, then $s_1 \neq s_n$.

    e). None of the above.

13. As discussed in class, we define $\mathbf{lcs}(A, B)$ as the *length* of the longest common subsequence of $A = a_1 a_2 \cdots a_n$ and $B = b_1 b_2 \cdots b_m$. For two sequences A and B (with $n > 0$ and $m > 0$), which is **not** a possible answer of $\mathbf{lcs}(A, B)$?

    a). 0

    b). 1

    c). n

    d). m

    e). $n + m$

14. In class, we show how to compute the length of $\mathbf{lcs}(A, B)$, i.e., function `L(i,j)`, in $O(mn)$ time. What is the worst-case time complexity if we compute `L(i,j)` recursively *without* memoization?

    a). $O(m^2 n^2)$

    b). $\Omega(2^n)$

    c). $o(n^3)$

    d). $O(n \log m)$

    e). $\Theta(mn)$

15. In `Change-making` problem, our goal is to find the minimum number of coins denominations $d_1, d_2, \ldots, d_k$ that add up to n cents. The denominations are $\{5, 10, 20, 50, 100\}$ cents (Singapore coin denominations, third series (2013-present; note that 100 cents coin refer to the 1 dollar coin)[1]). Assume that we are only using coins for our (small) purchases in Singapore. Which of the following statements is **FALSE**?

    a). There is no solution if $n\%5 \neq 0$

    b). If $n = 115$ cents, we need at least 3 coins

    c). If $n = 115$ cents, there are 4 coins that add up to n cents

    d). If $n = 775$ cents, there are 9 coins that add up to n cents

    e). If $n = 775$ cents, we need at least 10 coins

B Essay (25 marks)

B.1 Light Bulbs (7 marks)

There are n light bulbs. You have a special tester which tests whether the bulbs are faulty. To operate this tester, you put a **non-empty set of bulbs** into the tester. The tester will report one of two possible outcomes:

1. *All* the bulbs are good, or
2. *At least one* bulb is faulty, but with a caveat:
   You will not have any additional information on **which bulb** is faulty

B.1.1 Come-up with a Valid Testing Strategy (4 marks)

For all the n bulbs, you need to determine whether they are faulty or not.
Design a valid testing strategy to be able to determine which bulbs are faulty.
Any strategy that works will be given 2 marks.
The optimal strategy that uses minimum number of testings in the worst-case is given full 4 marks.

B.1.2 Minimum Number of Testings in the Worst-Case (3 marks)

Find the minimum number of testings required in the worst-case.
Please give the exact answer in terms of n.
If need be, update your previous answer in B.1.1 to use the optimal strategy.

[1] https://www.mas.gov.sg/currency/circulation-currency/circulation-currency-coins

B.2 Graph Coloring (8 marks)

A graph $G = (V, E)$ is *bipartite* if its vertex set can be partitioned into two parts $V = X \cup Y$ such that the two endpoints of every edge $e \in E$ belong to different parts.

Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with an *arbitrary* set $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors. We emphasize that *different* vertices u and v *may have different* sets $L(u)$ and $L(v)$.

B.2.1 Design and Analysis (6 marks)

Design a randomized or deterministic algorithm that selects a color $\phi(v) \in L(v)$ for every vertex $v \in V$ such that the chosen colors form a *proper* coloring: $\phi(u) \neq \phi(v)$ for every edge $e = \{u, v\} \in E$.

Show that your algorithm computes a desired coloring in polynomial time with a probability of at least 1/2. You can assume that the bipartition $V = X \cup Y$ of the vertex set V is already given.

B.2.2 Special Case (2 marks)

Suppose you are told that *all* vertices have *the same set* of FOUR (4) colors, i.e., for every vertex $v \in V, L(v) = \{$'red', 'green', 'blue', 'black'$\}$. Solve the above problem *deterministically* in polynomial time for this special case. You can still assume that the bipartition $V = X \cup Y$ of the vertex set V is already given.

B.3 LinkedIn (10 marks)

Role play: You are a top student who already satisfied the graduation requirements from a certain university and is now planning on 'learning and expanding network' as much as possible in your upcoming last (optional) semester. Given:

- $1 \leq N$, the integer number of remaining courses in that university that you can take in your last (optional) semester,
- $1 \leq K$, the maximum amount of integer units that you can still take in your last (optional) semester based on that university's rule, and
- a list of N pairs (d_i, s_i) that describe the number of integer unit $1 \leq d_i \leq K$ units and the lecturer name (a short string between 1 to 10 lowercase alphabet characters) of course i (numbered from course 1 to course N).

You need to figure out which courses to take in your final semester in order to maximize the amount of units taken in your last (optional) semester (yes, you like to study so much), but without breaking the university's limit of K units in that last (optional) semester. However, you have an additional requirement: You do not want to take more than one course given by the same lecturer (as each new lecturer that you study with is a new potential connection on LinkedIn![2]).

[2] You are welcome to add our LinkedIn profiles at `https://www.linkedin.com/in/sss1213/` and `https://www.linkedin.com/in/steven7halim/`

B.3.1 Subtask 1: At Most Two Different Lecturers (2 marks)

Suppose that of the N remaining courses, there are **at most two** different (challenging) lecturers left. For example $N = 3$ and $K = 10$ and the 3 courses left are: {(1, 'chang'), (2, 'halim'), (5, 'chang')}, then the optimal strategy is take the second course taught by (Prof Steven) Halim for 2 units and the third course taught by (Prof) Chang (Yi-Jun) for another 5 units. This way, you can study 7 more units in your final semester, which is not more than $K = 10$ units and without taking more than one course with the same lecturer.

Design a $\Theta(N^2)$-time complete search/brute force/try all algorithm that can correctly solve this subtask 1. There is no need to supply proof of correctness as long as your complete search is trying all possibilities.

B.3.2 Subtask 2: No two courses have the same lecturer (3 marks)

Suppose that of the N remaining courses, no two courses have the same lecturer. For example $N = 3$ and $K = 10$ and the 3 courses left are: {(7, 'diptarka'), (2, 'halim), (5, 'chang')}, then the optimal strategy is take (Prof) Diptarka (Chakraborty)'s and (Prof Steven) Halim's courses for a total of $7 + 2 = 9$ units. There is no other better combination.

Use an algorithm that you have learned in class *during lecture on Week 06* (i.e., Fibonacci, LCS, Knapsack, or Change-making) to correctly solve this subtask 2. There is no need to supply proof of correctness (as we have done so in class) as long as you can show how to use that algorithm correctly. Analyze the Big O time complexity in terms of N and/or K.

B.3.3 Manually Solve the Full Problem (1 mark)

For example $N = 4$ and $K = 10$ and the 4 courses left are: {(1, 'chang'), (7, 'diptarka'), (2, 'halim), (5, 'chang')}, then the optimal strategy is take (Prof) Chang (Yi-Jun)'s 1 unit course (avoid the 5 units course), (Prof) Diptarka (Chakraborty)'s, and (Prof Steven) Halim's courses for a total of $1 + 7 + 2 = 10$ units. This is the optimal answer.

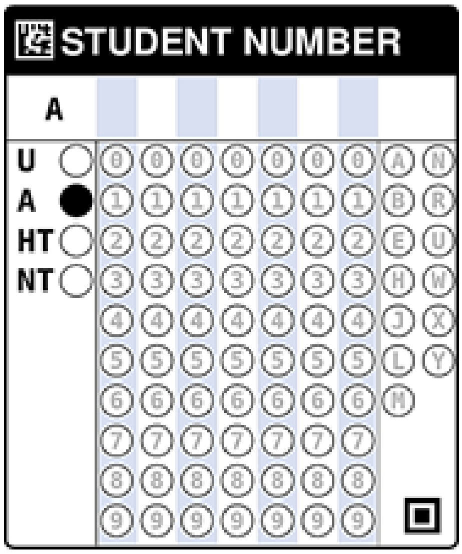
To ascertain that you have fully understand the full version of this problem, give one integer and a short explanation on how to get that answer, that describes the optimal output for the following test case: $N = 8$, $K = 100$, and the 8 courses are: {(79, 'p'), (43, 'v'), (94, 'a'), (91, 'c'), (25, 'p'), (31, 'a'), (70, 'a'), (12, 'v')}. That's it, there are 4 lecturers {'p', 'v', 'a', 'c'}. You will get partial marks if your answer is within 5 units away of the optimal answer.

B.3.4 Solve the Full Problem in $o(N^2 \cdot K)$ (4 marks)

Design a Dynamic Programming (DP) algorithm (just describe the base case(s) and the inductive step) to fully solve this problem and analyze its Big O time complexity in terms of N and/or K. In order to get full marks, your DP algorithm must run in polynomial time and in $o(N^2 \cdot K)$.

The Answer Sheet

Write your Student Number in the box below using **(2B) pencil**.
Do NOT write your name.



Write your MCQ answers in the special MCQ answer box below for automatic grading.
We do not manually check your answer.
Shade your answer properly (use (2B) pencil, fully enclose the circle; select just one circle).

No	1	2	3	4	5	6	7	8	9	10
A	○	○	○	○	○	○	○	○	○	○
B	○	○	○	○	○	○	○	○	○	○
C	○	○	○	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○	○	○	○
E	○	○	○	○	○	○	○	○	○	○

No	11	12	13	14	15
A	○	○	○	○	○
B	○	○	○	○	○
C	○	○	○	○	○
D	○	○	○	○	○
E	○	○	○	○	○

Box B.1.1. Come-up with a valid testing strategy

Box B.1.2. Minimum number of testings in the worst-case

Box B.2.1. Design and Analysis

Box B.2.2. Special case: all vertices have the same set of FOUR (4) colors

Box B.3.1. Subtask 1: At most two different lecturers

Box B.3.2. Subtask 2: No two courses have the same lecturer

Box B.3.3. Manually solve the full problem

Box B.3.4. Solve the full problem in $o(N^2 \cdot K)$

– END OF PAPER; All the Best –