LE/EECS 3101
Design & Analysis of Algorithms

Sample Final Exam

Shahin Kamalli

York University

15 December 2023 at 9:00

Write your name and student id here:

"Rivers know this: there is no hurry. We shall get there some day ... " *A. A. Milne, Winnie the Pooh*

- **Do not open this booklet until instructed**.
- You are NOT allowed to use any printed/written material. A "**cheat page**" is provided at the end of the exam.
- It is OK to take the staples off, but make sure to **submit all sheets**, including the cheat sheet.
- Please **turn off your cell phones** and put them in your bags.
- Calculators are not needed but you can use simple calculators with no memory.
- Manage your time. We start the exam at 9:00 and end the exam at 12:00. **You have 180 minutes**. Don't waste too much time on a single question. It is a long exam, and your time is limited. Many questions ask for a final answer, and you do not need to provide explanations. Use blank pages at the end of the exam if you need more space.
- To save trees, the exam is printed **double-sided**. There are **20 pages** in the final exam, including this cover page, the cheat page, and **four extra blank pages** (use them if you need more space for draft work or your final answers).
- There will be one or two optional bonus questions at the end of the exam. These questions are more complex and will be marked relatively harshly. Approach them only if you have finished other questions.
- The marks will be scaled so that the highest mark gets the full mark.

1. Short Answer Questions

Provide your short answers in the provided boxes. **There is no need to justify your answers.**
In the actual exam, there will be more short-answer questions. They cover all topics discussed in the course.

(a). True or False: $n \log n \in \Theta(n \log n^3 + \sqrt{n})$.

(b). True or False: Quick-Sort running time in expected to be $\Theta(n \log n)$, assuming that the input is permuted randomly.

(c). What is the asymptotic running time of the following pseudocode? Write your answer using Θ notation.

```
foo(n)
    for i ← 1 to n do {
        k ← i^5
        while k > 1 do
            k ← k/5
    }
    return i
```

(d). Assume $T(1) = 2022$ and $T(n) = 81T(n/3) + n^4 \log n$. Write down the asymptotic value of T using Θ notation. Hint: $\log_3 81 = 4$ and $\log_{81} 3 = 1/4$.

(e). From the set {Merge-Sort, QuickSort (with the pivot selected by the media-of-five algorithm), Counting-Sort, Bucket-Sort}, specify the most suitable sorting algorithm for the following scenario.
The input is a large array of n positive integers in the range $[1, 100]$. The input distribution is unknown.

(f). True or False: Adjacency matrix and adjacency list take asymptotically equal space of $\Theta(n^2)$ to store a graph of n vertices in which every vertex has $\Theta(n)$ neighbors.

(g). True or False: If one can find a polynomial-time algorithm for one NP-complete problem, then all NP-complete problems can be solved in polynomial time.

2. Time Analysis

(a) Provide a complete proof of the following statement from first principles (i.e., using the original definitions of order notation).

$$n^3 \in \omega(n^2(\log n)^2)$$

(b) Given a set of points on the x-axis (each point indicated by its distance from the origin), describe a Divide-and-Conquer algorithm that reports the closest pair of points! Assume the points are sorted by their x-coordinate. Write down the recursive formula for the running time and express it in Θ notation, using the Master theorem.

3. Heaps and Huffman (Short Answers)

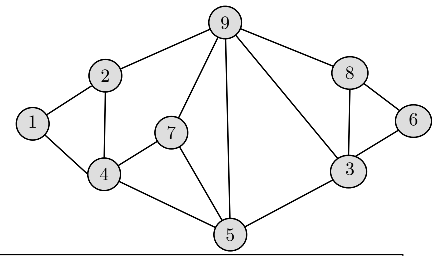
(a) Given the array $[1, 2, 3, 4, 5, 6, 7, 8]$, first Heapify the array to form a Max-Heap (using the linear-time Heapify algorithm), and then apply the Insert(10) operation on the resulting heap. Show the final heap after these operations. **There is no need to explain your answer or show intermediate steps.**

(b) Apply the Huffman encoding to encode the following string over alphabet $\Sigma = \{e, g, h, o, r, s, \_\}$. Break ties arbitrarily, and assume the tree with a smaller frequency forms the left child of a merged tree (and in case of equal frequencies, the subtree whose left-most leaf appears earlier in the alphabet forms the left child).
It suffices to show the (final) Huffman tree.

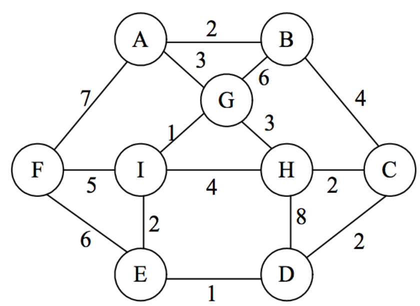
$$S = shoor\_eshgh$$

4. Graph Algorithms (Short Answers)

(a). Write down the vertex ordering by the DFS and BFS traversals (each in one line) of the following graph. Assume the source is 5 and G is represented with a sorted adjacency list.
There is no need to explain or justify your answers.



(b). Write down the first five edges added by Kruskal's and Prim's MST algorithm (each in one line) of the following graph. Break ties arbitrarily. Indicate each edge with its endpoints, e.g., (A, B).
There is no need to explain or justify your answers.



5. Selection

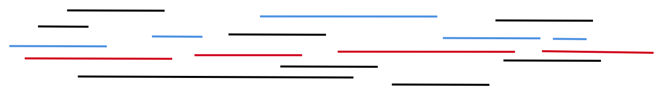
When doing Quick-Select, it is desired to have a *good* pivot which is almost in the middle of the sorted array. When doing the average-case analysis of Quick-Select, we considered a *good* and a *bad* case; the good case happened when the pivot was among the half middle items of the sorted array, i.e., we had $n/4 \leq i < 3n/4$ (i is the index of pivot in the partitioned array). Change the definition of the good case and assume the good case happens when we have $n/3 \leq i < 4n/5$. Provide an upper bound for $T(n)$ and specify whether this analysis shows that Quick-Select runs in $O(n)$.

6. Decision Trees

Use the Decision-Tree approach to prove a time complexity lower bound for sorting an array A of distinct integers that is already sorted except that two elements are misplaced, e.g., $A = [1, 2, 3, 4, \mathbf{9}, 6, 7, 8, \mathbf{5}, 10, 11, 12]$.

7. Dynamic Programming

Given a set of activities, each represented by a black interval, we would like to color each interval red, blue, or leave it black in a way that no two red intervals overlap, and similarly no two blue intervals overlap. See the figure below for an example. We would like to color intervals in a way to maximize the number of intervals colored red or blue.



(a) Devise a DP program to find the maximum number of intervals that can be colored red or blue. It suffices to show Steps 1 and 2 in the DP paradigm in your solution; that is, specify the subproblems and write down a recursive formula. Assume indices start at 1 and the intervals are sorted by their completion time.

(b) Describe a greedy algorithm for this variant of the interval-selection problem (in a few sentences or with pseudocode) and indicate whether this algorithm yields to an optimal solution.
There is no need to justify your answer.

8. Dynamic Programming

Given two strings S of length n and T of length m, we want to *reconfigure* S to T with the following operations: i) delete a character from S **at a cost of 2**, ii) insert a character into S **at cost of 2**, or iii) substitute a character in S into a new character **at cost of 3**. For example, to reconfigure $S = GoodMajor$ to $T = GoofyMayor$, we can substitute d with f (at cost 3), add y (at cost 2) and substitute y with j (cost 3) for a total cost of 8.

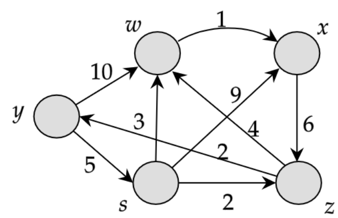
Devise a DP solution to report the minimum cost to reconfigure S to T . It suffices to show Steps 1 and 2 in the DP paradigm in your solution; that is, specify the subproblems and write down a recursive formula. Assume indices start at 1.

9. Knapsack Problem

Consider a variant of the knapsack problem in which you can accept, reject, or half-accept each item. As before, the input is a set of n items, where the i'th item has size s_i and value v_i. In case you half-accept an item v_i, a fraction $1/2$ of the item with size $s_i/2$ and value $v_i/2$ will be placed in the knapsack. Derive a Dynamic Programming solution that finds the value of the optimal solution. It suffices to define subproblems and provide a recursive formula for the value of optimal solutions.

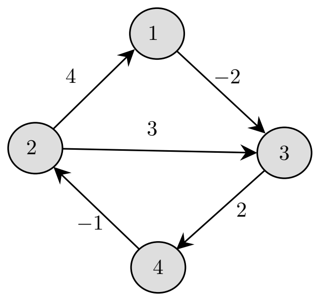
10. Dijkstra's Algorithms

Follow the steps of Dijkstra's algorithm on the following directed graph using vertex s as the source. Show the d and π values after each pass (after processing each vertex). For the π values, it suffices to shade or colour the edges from predecessors to nodes.



11. Floyd-Warshall's Algorithms

We have followed the steps of the Floyd-Warshall algorithm for finding the weight of all-pair shortest paths in the following graph. Most values in matrices D_i are provided for $i \in \{0, 4\}$. A few indices, however, are missing, and you should calculate them. It suffices to write down the missing values in the tables. **There is no need to justify your answer.**



	1	2	3	4
1	0	∞	-2	∞
2	4	0	3	∞
3	∞	∞	0	2
4	∞	-1	∞	0

D^0

	1	2	3	4
1	0	∞	-2	∞
2	4	0	2	∞
3	∞	∞	0	2
4	∞	-1	∞	0

D^1

	1	2	3	4
1	0	∞	-2	∞
2	4	0	2	∞
3	∞	∞	0	2
4	3	-1	1	0

D^2

	1	2	3	4
1	0	∞	-2	
2	4	0	2	
3	∞	∞	0	
4				

D^3

	1	2	3	4
1				
2				
3				
4				

D^4

The following space is intentionally left blank. Use it if you need more space for your answers or draft work.
Remember to submit this page.

You can consult the following "cheat page" in answering your questions. Not all material presented here is necessary to answer the exam questions. Remember to submit this page.

$$x^a x^b = x^{a+b}$$
$$(x^a)^b = x^{ab}$$
$$x^n + x^n = 2x^n \neq x^{2n}$$
$$x^n * x^n = x^{2n}$$
$$x^0 = 1$$
$$x^{-a} = \frac{1}{x^a}$$
$$\frac{x^a}{x^b} = x^{a-b}$$
$$(xy)^a = x^a y^a$$
$$2^n + 2^n = 2^{n+1}$$

$$\log_b(M) = X \leftrightarrow b^X = M$$
$$\log_b(b) = 1$$
$$\log_b(1) = 0$$
$$\log_b(0) \to \text{not defined}$$
$$\log_b(b^k) = k$$
$$b \log_b(M) = M$$
$$\text{if} \log_b(M) = \log_b(N), \text{ then } M = N$$
$$\log_b(M^k) = k \cdot \log_b(M)$$
$$b^{\log_b(M)} = M$$
$$\log_b\left(\frac{1}{M}\right) = \log_b(M^{-1}) = -1 \cdot \log_b(M)$$
$$\log_b\left(\frac{M}{N}\right) = \log_b(M) - \log_b(N)$$
$$\log_b(M \cdot N) = \log_b(M) + \log_b(N)$$
$$\log_b(M) = \frac{\log_c(M)}{\log_c(b)}$$

$$a^{\log_b n} = n^{\log_b a}$$

$$\sum_{i=1}^{n} 1 = n$$
$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$$
$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3$$
$$\sum_{i=1}^{n} i^k = 1^k + 2^k + \cdots + n^k = \frac{1}{k+1}n^{k+1}$$
$$\sum_{i=0}^{n} a^i = 1 + a + \cdots + a^n = \frac{a^{n+1}-1}{a-1}(a \neq 1); \quad \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$
$$\sum_{i=1}^{n} i2^i = 1 \cdot 2 + 2 \cdot 2^2 + \cdots + n \cdot 2^n = (n-1)2^{n+1} + 2$$
$$\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma, \text{ where } \gamma \approx 0.5772\ldots \text{ (Euler's Constant)}$$
$$\sum_{i=1}^{n} \lg i \approx n \lg n$$

$$T(n) = \begin{cases} a\,T\left(\frac{n}{b}\right) + f(n) & \text{if } n > 1 \\ d & \text{if } n = 1. \end{cases}$$

($a \geq 1$, $b > 1$, and $f(n) > 0$)

- Compare $f(n)$ and $n^{\log_b a}$
- Case 1: if $f(n) \in O(n^{\log_b a - \epsilon})$, then $T(n) \in \Theta(n^{\log_b a})$
- Case 2: if $f(n) \in \Theta(n^{\log_b a}(\log n)^k)$ for some non-negative k then $T(n) \in \Theta(f(n) \log n) = \Theta(n^{\log_b a}(\log n)^{k+1})$
- Case 3: if $f(n) \in \Omega(n^{\log_b a + \epsilon})$ and if $af(n/b) \leq cf(n)$ for **some constant** $c < 1$ (regularity condition), then $T(n) \in \Theta(f(n))$

The following space is intentionally left blank. Use it if you need more space for your answers or draft work. Remember to submit this page.

The following space is intentionally left blank. Use it if you need more space for your answers or draft work. Remember to submit this page.

The following space is intentionally left blank. Use it if you need more space for your answers or draft work.
Remember to submit this page.