Basic Algorithms 5/12/2025
Computer Science Department, New York University CSCI-UA.0310-007
Prof. Rotem Oshman Final Exam

NetID:	**Name:**

Do not open this booklet until you are directed to do so. Read all the instructions on this page.

Instructions

- **When the exam begins, write your name on every page of this booklet.**
- **Answer each question in the provided box. If you need more space, use the extra pages at the end of the booklet, and clearly indicate it in the original question's box. Do not use the back side.**
- **The exam is 110 minutes long. The maximum possible score is 100 points.**
- **Grading is based on correctness, clarity, and conciseness.**
- **You may use any algorithm or theorem we saw in class without proof, as long as you state it correctly. For all questions asking you to give algorithms, you do *not* have to give detailed pseudo-code, or even any pseudo-code. It is enough to give a clear description of your algorithm.**
- **This exam is closed book. No calculators, phones, smartwatches, etc. are permitted.**
- **Do not spend too much time on any one problem. Read them all through first, and attack them in the order that allows you to make the most progress.**
- **Good luck!!**

Name:

1. **(15 Points)** In this question there is no need to justify your answers.

   (a) (5 points) Check all correct answer(s). **There may be multiple correct answers.**
   If $f(n) = 2n$ and $g(n) = 2^n/2$:

   Solution: ☐ $f = \Omega(g)$ ☐ $f = O(g)$ ☐ $f = \Theta(g)$

   (b) (5 points) Check all correct answer(s). **There may be multiple correct answers.**
   If $f(n) = 2n^3 \log n$ and $g(n) = 100\,(n^3 \log n + n^2)$:

   Solution: ☐ $f = \Omega(g)$ ☐ $f = O(g)$ ☐ $f = \Theta(g)$

   (c) (5 points) Give a Θ-expression for the solution of the following recurrence:

   $$T(n) = T(n/2) + 5n$$

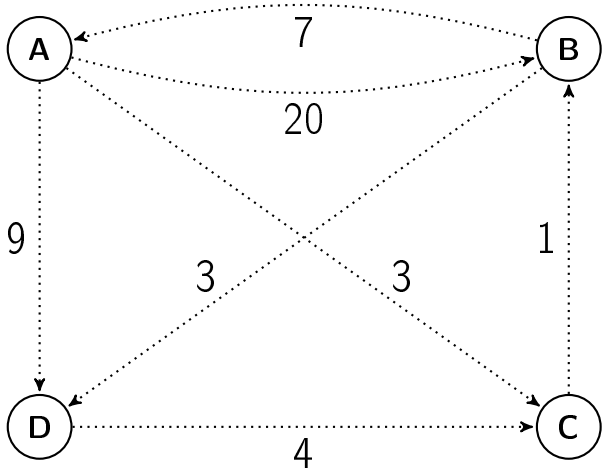
   Assume that $T(1) = 1$, and that n is a power of 2.

   Solution: $T(n) = \Theta($_____$)$

Name:

2. **(10 Points) Dijkstra's Algorithm**

Consider the following weighted directed graph $G = (V, E)$:



Run Dijkstra's algorithm with source vertex A on G. Assume that both the vertices and the adjacency lists are traversed in *alphabetic order*, i.e., when looping overall all vertices they are ordered as (A, B, C, D), and when looping over the adjacency list of a vertex its neighbors are ordered alphabetically.

For each vertex v, fill all the values that $d[v]$ takes during the course of the execution of Dijkstra, in the order in which these values appear. Note that the first value is ∞ for all vertices but A. Write only changes to a value, not repetitions of the same value (e.g., if $d[B] = \infty$ and after the first step of Dijkstra we still have $d[B] = \infty$, do not write the value ∞ twice.) Again, mind the alphabetic ordering.

	$d[v]$
A	_____
B	∞, _____
C	∞, _____
D	∞, _____

Name:

3. **(12 points) Messy Kruskal**

Suppose we run Kruskal's algorithm on a connected, undirected graph with non-negative edge weights, but we do not sort the edges in order of increasing weight, and instead we simply carry out the algorithm using whatever the order the edges are given to us in.

Which statements are necessarily true about the graph that this version of Kruskal outputs? Justify each of your answers by explaining why the statement is true, or giving a counter-example if it is not necessarily true.

(a) (4 points) The output will not contain any cycles.

Solution: ☐ True ☐ False
Justification:

(b) (4 points) The output will be connected.

Solution: ☐ True ☐ False
Justification:

Name:

(c) (4 points) If an edge e is included in the output, then there is some cut $(S, V \setminus S)$ in the graph G such that edge e is a light edge in the cut $(S, V \setminus S)$.

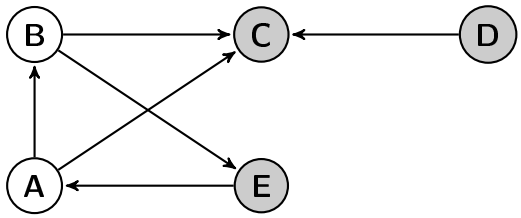
Solution: ☐ True ☐ False
Justification:

Name:

4. **(24 points) Graphs with Marked Vertices**

We are given a directed graph $G = (V, E)$, and a marking function $m : V \to \{0, 1\}$ that indicates whether a vertex $v \in V$ is *marked*, $m(v) = 1$, or *unmarked*, $m(v) = 0$.

For example, in the following graph, vertices C, D, E are marked (shown in gray), and the marking function is given by $m(A) = 0, m(B) = 0, m(C) = 1, m(D) = 1, m(E) = 1$.



In the following, you do not have to write pseudocode, but do not omit any details in your algorithm descriptions. If you are modifying an algorithm we saw in class, explain all the modifications and how they should be implemented. There is no need to provide justification.

(a) (8 points) Given a source vertex $s \in V$, describe an algorithm that outputs the number of marked vertices that are reachable from s.

For full credit, your algorithm should run in time $O(|V| + |E|)$.

(In the example above, if $s = A$, then the correct output is 2.)

Name:

(b) (8 points) Given a source vertex $s \in V$, describe an algorithm that outputs the set of vertices u that can be reached from s by a path that does not include any marked vertex, except possibly for vertices s and u themselves (which may or may not be marked).

For full credit, your algorithm should run in time $O(|V| + |E|)$.

(In the example above, if $s = B$, then the correct output is B, C, E.)

(c) (8 points) Assume that the graph G contains *exactly one cycle*, but we do not know what the cycle is (it is not given as part of the input). You do not need to check that the graph has exactly one cycle, this is given. Describe an algorithm that outputs "yes" if the cycle includes exactly one marked vertex, and "no" otherwise.

For full credit, your algorithm should run in time $O(|V| + |E|)$.

(In the example above, the output should be "yes": the graph contains exactly one cycle, $A \to B \to E \to A$, and this cycle has exactly one marked vertex, E.)

Name:

5. **(14 Points) Subway Routes (Dynamic Programming)**

Consider a simplified subway map of NYC, represented by an array $S[1, \ldots, n]$ of positive numbers. Each index $i = 1, \ldots, n$ represents a subway stop where we can take the subway from position i to position $i + S[i]$.

A person starts in position 1, and wants to reach position n. At each position i, the person can do one of the following:

- Walk on foot to position $i + 1$. This takes one time unit.
- Take the subway from position i to position $i + S[i]$. This also takes one time unit.

You may assume that $i + S[i] \leq n$ for all i (that is, the subway never goes past position n).

Example

For $n = 5$ and $S = [2, 3, 2, 1, 1]$, two possible routes from position 1 to position $n = 5$ are:

- Start at position 1, take the subway to position 3, walk to position 4, take the subway to position 5. This route takes 3 time units.
- Start at position 1, walk to position 2, take the subway to position 5. This route takes 2 time units.

In this example, the shortest time required to go from position 1 to position $n = 5$ is 2 time units (the second route above is the shortest).

Let $T[i]$ denote the shortest amount of time required to reach position n starting from position i, for each $i = 1, \ldots, n$.

(a) (6 points) Complete the following recursive formula for $T[i]$. (Hint: think carefully about which values $T[j]$ are needed to compute $T[i]$.)

$$T[i] = \begin{cases} \underline{\qquad\qquad\qquad\qquad}, & \text{if } i \ \underline{\qquad\qquad} \\ \underline{\qquad\qquad\qquad\qquad}, & \text{if } i \ \underline{\qquad\qquad} \end{cases}$$

Name:

(b) (5 points) Using the previous part, write a dynamic programming algorithm that outputs the shortest amount of time required to go from position 1 to position n.

(c) (3 points) Specify the running time of your dynamic programming algorithm from part (b), as a function of n.

$$\Theta\Big(\underline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}\Big)$$

Name:

6. **(25 Points) Short Answers**

Each multiple-choice question below has exactly one correct answer. Clearly write the correct answer **in the box**. You do **not** have to justify your answers.

(a) (5 points) Using the sorting algorithms that we learned in class, the fastest time possible to sort an array of n numbers in the range $\{1, \ldots, 2n\}$ is:

(A) $\Theta(n^2)$
(B) $\Theta(n \log n)$
(C) $\Theta(n)$
(D) None of the above

(b) (5 points) Recall that the Floyd-Warshall dynamic programming algorithm computes shortest path distances between all pairs of vertices in $n = |V|$ iterations, by filling in a three-dimensional table $F[u, v, i]$, where u is a source vertex, v is a destination vertex, and i is the iteration number. (Alternatively, we can think of this as filling in n two-dimensional tables, one for each iteration.) After all n iterations, $F[u, v, n]$ contains the shortest-path distance from u to v for every $u, v \in V$.

Let $\text{dist}(u, v)$ denote the true distance from vertex u to vertex v. Which of the following statements is always true of cell $F[u, v, i]$ where $i < n$ (i.e., not the last iteration)?

(A) $F[u, v, i] \geq \text{dist}(u, v)$.
(B) $F[u, v, i] \leq \text{dist}(u, v)$.
(C) None of the above.

(c) (5 points) When we perform DFS, if $(u, v) \in E$ and $d[u] < d[v]$ (that is, the discovery time of vertex u is smaller than the discovery time of vertex v), then it is necessarily true that:

(A) (u, v) is not a tree edge.
(B) (u, v) is not a forward edge.
(C) (u, v) is not a cross edge.
(D) None of the above.

(d) (5 points) In the Gale-Shapley algorithm for computing a stable matching, if hospital h is matched with resident r in the output, which of the following is necessarily true?

(A) Hospital h ranks resident r above all other residents.
(B) Resident r ranks hospital h above all other hospitals.
(C) None of the above.

Name:

(e) (5 points) Are the following problems in NP?

- P_1: given a bipartite graph $G = (L, R, E)$ and a number k, find a matching $M \subseteq E$ of size at least k, or say that there is none.
- P_2: given a graph $G = (V, E)$ and a number k, find a simple path of length at least k in G, or say that there is none.

(A) $\mathsf{P}_1 \in \mathsf{NP}$ and $\mathsf{P}_2 \in \mathsf{NP}$.
(B) $\mathsf{P}_1 \notin \mathsf{NP}$ and $\mathsf{P}_2 \in \mathsf{NP}$.
(C) $\mathsf{P}_1 \in \mathsf{NP}$ and $\mathsf{P}_2 \notin \mathsf{NP}$.
(D) $\mathsf{P}_1 \notin \mathsf{NP}$ and $\mathsf{P}_2 \notin \mathsf{NP}$.

Name (no need to fill in if you do not use this page):

Extra sheet of paper #1. If you use this sheet make a note of it on the original question.

Solution:
Question number:

Name (no need to fill in if you do not use this page):

Extra sheet of paper #2. If you use this sheet make a note of it on the original question.

Solution:
Question number:

Scratch paper: anything written here will NOT be graded!

Scratch paper: anything written here will NOT be graded!

Cheat Sheet

Logarithm Rules

The following holds for any basis b (and c) and values X, Y.

$$\begin{aligned}
\log_b(b^X) &= X \\
b^{\log_b(X)} &= X \\
\log_b(X \cdot Y) &= \log_b(X) + \log_b(Y) \\
\log_b\left(\frac{X}{Y}\right) &= \log_b(X) - \log_b(Y) \\
\log_b(X^Y) &= Y \cdot \log_b(X) \\
\log_b(X) &= \frac{\log_c(X)}{\log_c(b)}
\end{aligned}$$

Arithmetic Series

Let d be a constant such that $a_n := a_{n-1} + d$ for all $n \geq 1$ and a_0 some initial value. We have

$$a_0 + a_1 + \ldots a_k = \sum_{i=0}^{k} a_i = \frac{k+1}{2}(a_0 + a_k).$$

Geometric Series

For a *finite* geometric series we have

$$a + a \cdot r + a \cdot r^2 + \cdots + a \cdot r^k = \sum_{i=0}^{k} a \cdot r^i = \frac{a \cdot (1 - r^{n+1})}{1 - r}$$

and in particular

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1 = \Theta(2^k).$$

If $|r| < 1$, then the *infinite* geometric series converges:

$$a + a \cdot r + a \cdot r^2 + \cdots = \sum_{i=0}^{\infty} a \cdot r^i = \frac{a}{1 - r}$$

and in particular

$$\sum_{i=0}^{\infty} \frac{1}{2^i} = 2 = \Theta(1).$$