Basic Algorithms (CSCI-UA.0310-001/003) Practice Final **Instructor: Professor Oded Regev**

Computer Science Department, New York University

NetID:	**Name:**

Instructions (Read every item carefully!)

- **Do not open this booklet until you are directed to do so. Read all the instructions on this page.**
- **When the exam begins, write your name on every page of this quiz booklet.**
- **Answer each question in the provided box. If you need more space, write in the extra box towards the end of the booklet.**
- **The exam is 110 minutes long. The maximum possible score is 100 points.**
- **Grading is based on correctness, clarity, and conciseness.**
- **If you don't know how to answer a question, please say so, and explain what you tried. You will get significantly more points doing so.**
- **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. **(20 Points)**

   (a) State if $f = \Omega(g)$, if $f = O(g)$ or if $f = \Theta(g)$. Give a short justification. $f(n) = n^2 + n$ and $g(n) = 100n^{3/2}$.

   (b) State if $f = \Omega(g)$, if $f = O(g)$ or if $f = \Theta(g)$. Give a short justification. $f(n) = 4^{\sqrt{n}}$ and $g(n) = 2^n$.

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

   $$T(n) = 2T(\lceil n/3 \rceil) + n \; .$$

   Assume $T(1) = T(2) = 1$. Briefly justify your solution.

   (d) Give a Θ expression for the solution of the following recurrence:

   $$T(n) = 3T(\lceil n/2 \rceil) + n \; .$$

   Assume $T(1) = 1$. Briefly justify your solution.

2. **(20 Points)**

(a) Consider a file that uses the following list of symbols with the following frequencies:

Letter	A	B	C	D	E	F	G
Frequency	0.02	0.07	0.14	0.15	0.17	0.18	0.27

Find an optimal prefix code based on Huffman's algorithm. Describe both the code (i.e., mapping from symbols to bit strings) and the corresponding tree.

(b) Show the running of Dijkstra's algorithm for finding the distance from A to all other vertices in the graph from Figure 1. You should show the updated distances to all the vertices (i.e., the key of the vertex in the priority queue) after each time a vertex is extracted from the priority queue in the algorithm. (Fill in the following table with the distances from A to all other vertices. Each row represents the distances after one more vertex is popped from the queue.)

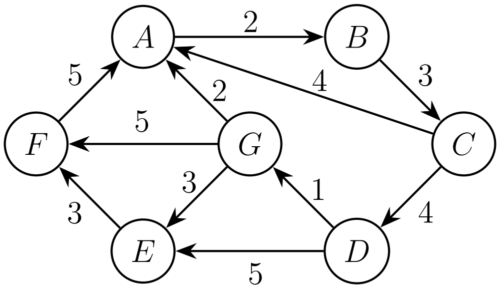

Figure 1: A directed weighted graph G.

Key in Queue:	A	B	C	D	E	F	G
Step 1:	0	∞	∞	∞	∞	∞	∞
Step 2:	0						
Step 3:	0						
Step 4:	0						
Step 5:	0						
Step 6:	0						
Step 7:	0						

3. **(20 Points)** Given an input string of n characters $s[1 \dots n]$ and a set of valid words D, design an algorithm that determines if the input string can be segmented into a space-separated sequence of valid words.

   For example, if you are given $D = \{$am, are, i, important, is, you$\}$ and an input string "iamimportant", you can split the string as "i am important". Thus for this input your algorithm should return True. Assuming checking containment in D takes only constant time, your algorithm should run in $O(n^2)$ time. Justify the correctness and run-time bound of your algorithm.

4. **(20 Points)** Consider a directed graph where each edge is colored in either red or blue. Describe an $O(|V| + |E|)$ time algorithm that given vertices u and v, determines if there exists a path from u to v that uses **at least one** blue edge. The path does not need to be simple (i.e., you can have repeated vertices and edges in the path). Prove the correctness of your solution. (Hint: Reduce the problem to checking if a path exists between two vertices in a directed graph. Recall the separable path question in the homework; a similar idea will work here.)

5. **(20 Points)** Consider a connected undirected graph where each edge has an integer weight as well as a color which is either red or blue. Give an efficient algorithm to find an MST of the graph with the smallest number of red edges. In other words, among all possible MSTs, the algorithm should output one that has the least number of red edges. Your algorithm should run in time $O(|E| \log |V|)$. Prove the correctness of your solution. (Hint: reduce the problem to a standard MST problem on weighted graphs without edge colors).

Name:________________________

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

Solution:
Question number:

Name:________________________

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

Solution:
Question number:

Name:________________________

Extra sheet of paper #3. If you use this sheet make a note of it on the original question.

Solution:
Question number:

Name:________________________

Scratch paper: anything written here will NOT be graded!

Name:________________________

Scratch paper: anything written here will NOT be graded!