Basic Algorithms (CSCI-UA.0310-005/6) Midterm **Instructor: Rachit Garg**
Computer Science Department, New York University

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 box towards the end of the booklet, and clearly indicate it in the original question's box. Do not use the back side.**
- **The exam is 75 minutes long. The maximum possible score is 100 points.**
- **Grading is based on correctness, clarity, and <u>conciseness</u>.**
- **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. **Asymptotics (30 Points)**

   Answer True or False for the following questions. If True, provide a short (one or two sentences) justification. If False, provide a counter-example (if possible) or a short justification.

   Hint: If $f = O(g)$, then $g = \Omega(f)$ and vice versa.

   (a) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.

   (b) If $f(n) = \Omega(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.

   (c) If $h(f(n)) = O(h(g(n)))$, then $f(n) = O(g(n))$.

   (d) If $T(1) = T(2) = 1$ and the following recurrence holds for $n > 2$:

   $$T(n) = T\left(\frac{n}{2}\right) + \sqrt{n}\ ,$$

   then $T(n) = \Theta(\log_2 n)$.

   Write the following functions in the Θ-notation in the following simple form $c^n \cdot n^d \cdot (\log_2 n)^r$, where c, d, r are constants. For example, $20n^2 + 5n + 7 \sim n^2$. (You can omit "Θ" in front of n^2.)
   You do **not** have to justify your answers.

   (d) $\frac{9n^4}{4} + 2n \sim$ ____________________

   (e) $\log_2(n^n + 2) \sim$ ____________________

2. **Divide and Conquer (30 points)**

Suppose you are given an array $A[1, \ldots, n]$ with n *distinct* integer elements in *sorted order*. We say that the element $A[k]$ is an **identity element** of A if $A[k] = k$.

(a) (5 points) Describe an $O(n)$ algorithm that finds an identity element. Note that there could be multiple identity elements in A. Your algorithm needs only to return (an arbitrary) one of the many identity elements. If A has no identity element, it should return **FALSE**. (No pseudo-code is needed.)

(b) (6 points) Suppose you look at $x = A[k]$. If $x < k$, then at most one of the following two array slices may contain an identity element:

(1) $A[1, \ldots, k-1]$

(2) $A[k+1, \ldots, n]$

Specify which slice contains the identity element, if there is any, by checking ✓ the corresponding box:

☐ $A[1, \ldots, k-1]$

☐ $A[k+1, \ldots, n]$

Justify your answer:

Name:________________________

(c) (8 points) Consider the following algorithm $\textsc{IdFind}(A, j, k)$ that finds an $i \in \{j \dots k\}$ such that $A[i] = i$, or returns **FALSE** if no such element i exists. Fill in the blanks (denoted ...) to complete the algorithm such that it runs in time sub-linear in $n := k - j + 1$.

```
IdFind(A, j, k)
    if j > k return FALSE
    Set i := ____________
    if A[i] = ____________ return ____________
    if A[i] < ____________ return IdFind(A, ____________, ____________)
    return IdFind(A, ____________, ____________)
```

(d) (6 points) Does the algorithm work if the elements of A are not distinct? If yes, prove so. Otherwise provide a counterexample.

(e) (5 points) Let $T(n)$ be the running time for your Divide-and-Conquer algorithm (from the previous part) on an input array of size n. Derive a recurrence relation for $T(n)$ and write a Θ-expression for $T(n)$.

3. **Dynamic Programming (40 points)**

   You are given an array $A[1, \ldots, n]$ of n non-negative integers and a target $T \geq 0$. Our goal is to determine whether there exists a subset of values of A that sums to T. In other words, we want to know whether there exist indices $\mathcal{I} \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in \mathcal{I}} A[i] = T$. (Note that the subset does not have to be consecutive, i.e., we can consider $A[1] + A[3] = T$.)

   For example, if $n = 5$ and $A = (3, 5, 2, 11, 3)$, then the answer is `true` for $T = 10$ (e.g., 3+5+2=10), but `false` for $T = 9$.

   (a) (6 points) How many distinct subsets are there in in array of length n? Briefly explain.

   (b) (5 points) Let $\mathsf{Subset}[i, t] :=$ `true` if the subarray $A[1, \ldots, i]$ contains a subset that sums to t and $\mathsf{Subset}[i, t] =$ `false`, otherwise. State appropriate base cases for $\mathsf{Subset}[0, t]$ (for $0 \leq t \leq T$).

Name:________________________

(c) (10 points) Derive a recursive formula for $\mathsf{Subset}[i, t]$ in terms of its subproblems. You may use logical operators such as $\wedge$ (logical and) and $\vee$ (logical or) in your formula.

(**Hint**: Note that a solution can choose each element of A at most once.)

(d) (8 points) To check your recursive formula, fill out the following table for Subset given the following input array: $A = [1, 3, 2]$ and $T = 5$.

(**Hint**: You should still be able to fill this table, even if you did not write the recursive formula.)

i \ t	0	1	2	3	4	5
0						
1						
2						
3						

(e) (6 points) Use the previous parts to give a Θ-expression for the running time of your algorithm to fill the array Subset. Justify your running time.

(f) (5 points) Assuming that the array Subset has been correctly populated, describe what value is returned to solve the problem of deciding whether there exists a subset that sums to T. Also, give a Θ-expression for the *total* running time of your algorithm and provide a short justification.

Name:________________________

Extra sheet of paper #1. If you use this sheet, you MUST make a note of it on the original question. OTHERWISE, THIS SHEET WILL NOT BE GRADED.

Solution:

Question number:

Name:________________________

Extra sheet of paper #2. If you use this sheet, you MUST make a note of it on the original question. OTHERWISE, THIS SHEET WILL NOT BE GRADED.

Solution:

Question number:

Name:________________________

Extra sheet of paper #3. If you use this sheet, you MUST make a note of it on the original question. OTHERWISE, THIS SHEET WILL NOT BE GRADED.

Solution: Question number:

Name:________________________

Scratch paper: anything written here will NOT be graded!

Name:________________________

Scratch paper: anything written here will NOT be graded!

Name:________________________

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

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