

Write your answers in the separate answer booklet.

1. **Multiple Choice:** Each question below has one of the following answers.

A: $\Theta(1)$ B: $\Theta(\log n)$ C: $\Theta(n)$ D: $\Theta(n \log n)$ E: $\Theta(n^2)$ X: I don't know.

For each question, write the letter that corresponds to your answer. You do not need to justify your answers. Each correct answer earns you 1 point. Each X earns you $\frac{1}{4}$ point. **Each incorrect answer costs you $\frac{1}{2}$ point.** Your total score will be rounded **down** to an integer. Negative scores will be rounded up to zero.

(a) What is $\sum_{i=1}^n \frac{n}{i}$?

(b) What is $\sum_{i=1}^{\lg n} 4^i$?

(c) How many bits are required to write $n!$ in binary?

(d) What is the solution of the recurrence $T(n) = 4T(n/2) + n \log n$?

(e) What is the solution of the recurrence $T(n) = T(n-3) + \frac{5}{n}$?

(f) What is the solution of the recurrence $T(n) = 9T(\lceil \frac{n+13}{3} \rceil) + 10n - 7\sqrt{n} - \frac{\lg^3 n}{\lg \lg n}$?

(g) How long does it search for a value in an n -node binary search tree?

(h) Given a sorted array $A[1..n]$, how long does it take to construct a binary search tree for the elements of A ?

(i) How long does it take to construct a Huffman code, given an array of n character frequencies as input?

(j) A train leaves Chicago at 8:00pm and travels south at 75 miles per hour. Another train leaves New Orleans at 1:00pm and travels north at 60 miles per hour. The conductors of both trains are playing a game of chess over the phone. After each player moves, the other player must move before his train has traveled five miles. How many moves do the two players make before their trains pass each other (somewhere near Memphis)?

2. Describe and analyze an algorithm to find the length of the longest substring that appears both forward and backward in an input string $T[1..n]$. The forward and backward substrings must not overlap. Here are several examples:

- Given the input string ALGORITHM, your algorithm should return 0.
- Given the input string RECURSION, your algorithm should return 1, for the substring R.
- Given the input string REDIVIDE, your algorithm should return 3, for the substring EDI. (The forward and backward substrings must not overlap!)
- Given the input string DYNAMICPROGRAMMINGMANYTIMES, your algorithm should return 4, for the substring YNAM.

For full credit, your algorithm should run in $O(n^2)$ time.

3. The *median* of a set of size n is its $\lceil n/2 \rceil$ th largest element, that is, the element that is as close as possible to the middle of the set in sorted order. It's quite easy to find the median of a set in $O(n \log n)$ time—just sort the set and look in the middle—but you (correctly!) think that you can do better.

During your lifelong quest for a faster median-finding algorithm, you meet and befriend the Near-Middle Fairy. Given any set X , the Near-Middle Fairy can find an element $m \in X$ that is *near* the middle of X in $O(1)$ time. Specifically, at least a third of the elements of X are smaller than m , and at least a third of the elements of X are larger than m .

Describe and analyze an algorithm to find the median of a set in $O(n)$ time if you are allowed to ask the Near-Middle Fairy for help. [*Hint: You may need the PARTITION subroutine from QUICKSORT.*]

4. SUBSETSUM and PARTITION are two closely related NP-hard problems.
- SUBSETSUM: Given a set X of integers and an integer k , does X have a subset whose elements sum up to k ?
 - PARTITION: Given a set X of integers and an integer k , can X be partitioned into two subsets whose sums are equal?
- (a) Describe and analyze a polynomial-time reduction from SUBSETSUM to PARTITION.
- (b) Describe and analyze a polynomial-time reduction from PARTITION to SUBSETSUM.
5. Describe and analyze efficient algorithms to solve the following problems:
- (a) Given a set of n integers, does it contain a pair of elements a, b such that $a + b = 0$?
- (b) Given a set of n integers, does it contain three elements a, b, c such that $a + b + c = 0$?