

MTH 4320 Homework 1

Yaohui Wu

February 5, 2024

Contents

1	Problem 1	1
2	Problem 2	1
3	Problem 3	1
4	Problem 4	2

1 Problem 1

Solution.

We have a triple nested loop and for the outermost loop we run $n - 1$ operations then we run $n - 1$ operations in the inner loop for every operation in the outer loop and so on. The running time is $O(n) \cdot O(n) \cdot O(n) = O(n^3)$. ■

2 Problem 2

Solution.

We have $n - 2$ operations from the outer loop where n is the input number *num*. Then we call the *is_prime* function twice for every operation and the function runs at most $n - 2$ operations every time. The running time is $2 \cdot O(n) \cdot O(n) = O(n^2)$. ■

3 Problem 3

Solution.

1. Store the first element of the input list L in a variable M .
2. For every element in L :
 - If it is greater than M then update M with that element.

3. We found the first largest element M in L .
4. Let N be the first element of L and $N \neq M$.
5. For every element in L :
 - If it is greater than N but less than M then update N with that element.
6. We found the second largest element N in L .
7. Let $M := N$ and N be the first element of L where $N \neq M$.
8. Repeat steps 2-6 until we found the 10th largest element M .

The running time is $10 \cdot O(n) + O(1) = O(n)$. ■

4 Problem 4

Solution.

1. Let M be the middle element of the list, L be the first element, and R be the last element.
2. If $M < x$ then we update L to the $M + 1$ element else we update R to the M element. Repeat until we find the leftmost x .
3. Similarly, we repeat steps 1-2 but we change the condition to $M \leq x$ to find the rightmost x .
4. The index of the rightmost x minus the index of the leftmost x is the number of times x appears in the list.

The running time is $2 \cdot O(\log n) + O(1) = O(\log n)$. ■