CP312 Algorithm Design and Analysis I **Winter 2024**

Assignment 2

Instructor: Dariush Ebrahimi *Due Date: 25-Feb-2024*

Instructions: You must submit your solutions as a single PDF file to MyLS. Make your solutions as detailed as possible by clearly stating every step in your answers. The assignment must be done individually. Any **COPYING** of solutions from external sources will result in a **ZERO** grade.

Problems

1. **Ternary Search.** Consider the following **ternary** search algorithm which requires us to search for an element k in a *sorted* array A of size n starting at index p and ending at index q. Unlike binary search, ternary search divides the array into 3 equally sized parts instead of two halves.

```
1: procedure TernarySearch(A, p, q, k)
2:     if p ≤ q then
3:         m1 = p + ⌊(q − p)/3⌋
4:         m2 = m1 + ⌊(q − p)/3⌋
5:         if A[m1] == k then
6:             return m1                          ▷ Found k at index m1
7:         end if
8:         if A[m2] == k then
9:             return m2                          ▷ Found k at index m2
10:        end if
11:        if k < A[m1] then
12:            return TernarySearch(A, p, m1 − 1, k)
13:        else if k > A[m2] then
14:            return TernarySearch(A, m2 + 1, q, k)
15:        else
16:            return TernarySearch(A, m1 + 1, m2 − 1, k)
17:        end if
18:    end if
19: end procedure
```

(a) (5 points) Write down the recurrence for this algorithm.

(b) (5 points) Solve the recurrence using any method you prefer.

2. **Average-Case Analysis.** Consider the deterministic (i.e. non-randomized) version of quicksort that we saw in class which takes the last element in the array as the pivot to partition the rest of the elements around.

(a) (5 points) In the best case, the pivot always splits the array in half for **all** recursive calls. Give an example of a sequence of 7 distinct numbers that cause this best-case behavior.

(b) (5 points) Suppose that all the elements in the array are equal. Write down the recurrence and solve it to find the running time of quicksort in this case.

3. **Tree-based Sorting.** The following is an array representation of an almost complete binary tree where node i's left child and right child are located at index $2i$ and $2i + 1$, respectively.

2	31	14	49	84	16	42	63

(a) (5 points) Draw the binary tree represented by this array.

(b) (6 points) Describe step-by-step how you will use HEAPIFY(i) to turn the tree into a heap (you may need to call HEAPIFY multiple times).

(c) (7 points) Describe how you will use this heap (and its operations) to get the same elements back into an array but in sorted order. State the running time of this procedure.

4. **Linear-Time Sorting.** You are given the following array consisting of 9elements.

$$A = (536, 822, 16, 012, 357, 210, 216, 316, 450)$$

(a) (6 points) Use Radix Sort to sort the sequence of elements in A. Show each step of your procedure.

(b) (6 points) Suppose that you modified Radix sort so that, instead of sorting from the least significant digit first (right-to-left), it would start sorting from the most significant digit (left-to-right). Show the output of this modified Radix sort algorithm when given A as input. Is it correct?