

# BRAC UNIVERSITY

Merul Badda, Dhaka, Bangladesh

## CSE221: Algorithms

### Assignment 1

Fall 2024

---

**1. Write the tight upper bound (big-O) for the following equations:**

- A.  $f_1(n) = n^3 + 7n^2$
- B.  $f_2(n) = (\log n)^{2023}$
- C.  $f_3(n) = \log(n!)$
- D.  $f_4(n) = n^2 \log_n(n^n)$
- E.  $f_5(n) = n * \sqrt[5]{n^2}$

**2. Find the time complexity of the following code snippets:**

- A. 

```
for (i = 1; i <= n; i++) {
    for (j = 1; j < i; j *= 2) {
        op();
    }
}
```
- B. 

```
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j += i) {
        op();
    }
}
```

**3. Express the following running time  $T(n)$  with tight upper bound (big-O) using the substitution method:**

- A.  $T(n) = 3T(\frac{n}{2}) + O(n)$
- B.  $T(n) = 8T(\frac{n}{4}) + O(n\sqrt{n})$

**4. Consider an array containing  $N$  unique values where, for some index  $i$ , the values are in decreasing order from index 0 to  $(i-1)$ , and then again from  $i$  to  $(N-1)$ . Moreover, it is guaranteed that the value of index 0 is smaller than the value of index  $(N-1)$ .**

An example array is given below:

index	0	1	2	3	4	5	6	7
value	5	4	1	12	10	9	7	6

Here  $i=3$ , it means the values are in decreasing order from index 0 to 2, and then again from 3 to 7. Also, the value from index 0 is smaller than the value of index 7.

**Given such an array, design an algorithm to find the index  $i$  with a better time complexity than  $O(N)$ .**

- i) Write the pseudocode of your designed algorithm.
- ii) Calculate the time complexity of your algorithm.

**5.** Applicable sorting algorithms: Selection, Bubble, Insertion, Merge, and Quick sort.

Scenario 1: You are to maintain an array in which new values are appended frequently. In addition, the array must be kept sorted in ascending order.

- a) In order to keep the array sorted, which of the applicable sorting algorithms will be the most efficient algorithm in terms of time complexity? Explain briefly.
- b) Simulate the algorithm on the array 1, 4, 7, 8, 11 when the next value to append is 3.

Scenario 2: You are to sort the following unsorted array of length  $N$  in ascending order. However, you are provided with a special processing unit that can only perform tasks with a time complexity of  $N \log N$ .

3	8	1	5	12	5	10	7	6
---	---	---	---	----	---	----	---	---

- a) Which of the applicable sorting algorithms will you use to sort the given array? Explain briefly.
- b) Simulate the algorithm on the given array.

**6.** Imagine a farmer named Sam who grows different types of vegetables in several continuous fields. Each field can grow a specific type of vegetable, and based on the market value of that vegetable, each field will bring a certain profit (positive integer) or loss (negative integer). The profit or loss is estimated and noted for each field.

Sam can start farming from any field, but once he starts, he must continue farming the next fields in sequence without skipping any, until he decides to stop, because his tractor can only move to the next adjacent field and cannot skip fields.

Sam needs to choose which sequence of fields to farm to maximize his profits. The following array represents the estimated profits or losses for ten consecutive fields on Sam's farm, based on the types of vegetables that can be grown in each field and their respective market values.

4	-18	16	-14	12	-1	3	1	-20	15
---	-----	----	-----	----	----	---	---	-----	----

Given the array with  $N$  values, you are to design an algorithm to decide which sequence of fields to farm for maximum profit. Your algorithm must not exceed the time complexity of  $O(N \log N)$ .

- a) Write the pseudocode of your designed algorithm and simulate the algorithm on the given array.
- b) Calculate the time complexity of your algorithm.