Assignment 2 Semester: Fall 2025
Deadline:12/11/25, 11:59 PM Full Marks: 30

CSE 221: Algorithms

1. Analyze the code snippet and form recurrence relation of the following function "func" and calculate the time complexity.

```
int func(int n) {
    if (n <= 1) {
        return 1;
    } else {
        int a = 0;
        for (int i = 0; i < 3; i++) {
            a += func(n / 2);
        }
        int b = func2(n);
        return a + b;
    }
}

int func2(int n) {
    int c = 0;
    for (int k = n; k >= 1; k -= 4) {
        for (int m = 0; m < n; m += 2) {
            c += k + m;
        }
    }
    return c;
}
```

2. Analyze the code snippet and form recurrence relation of the following function "func" and calculate the time complexity.

```
1.  Function F(n):
2.      If (n == 1):
3.          Return True;
4.      Else:
5.          X = F(n/2);
6.          Y = F(n/2);
7.          Z = F(n/2);
8.          U = A(n);
9.
10. Function A(n):
11.     For i := n to 1; i--:
12.         Return True;
```

4. Write the worst case time complexity of quick sort? Illustrate an array where the worst case of quick sort occurs if the first element is chosen as pivot.

5. A football team's analytics department hired you and gave you a task of reviewing match performance data from the past season. Each match's net goal difference (goals scored minus goals conceded) is recorded in a list. Now you were asked by the department to identify the team's best consecutive performance streak, where the net goal difference was maximized over a series of matches.
net_goal_difference: [1, 4, 3, -5 , 5, 6, 1, -4]

a. Solve the problem in O(n) time complexity. [
   Kadane's Algorithm in 2 minutes | LeetCode Pattern]

To prove your skills, you used an approach where the problem is divided into subproblems to find the solution. Now tell us more about it by answering the following questions.

i. For the given input above, determine the number of times the Cross-Sum exceeds both the Left-Sum and the Right-Sum. Exclude base cases (subarrays of size 1) from consideration.

Show your work properly.

Cross-Sum: sum of elements calculated by combining sum from mid to left and mid to right of the current subarray.

Left-Sum: maximum sum in the left half of the subarray.

Right-Sum: maximum sum in the right half of the subarray.

ii. The department wants you to find the maximum consecutive matches with a positive goal difference. Now present your idea to solve the problem considering you have to use the same approach used earlier with pseudocode/programmable code/step-by-step logical instructions.

6. What is a stable sorting algorithm and simulate sorting an array using quick sort to prove that it is a not stable sorting algorithm.

7. You are working on a grading system for a national standardized exam where students are marked out of 100. You need to sort millions of student scores as efficiently as possible.

Speed is critical, and memory is not a major constraint. Comparisons are costly.

Which algorithm is the most suitable for the above scenario? Explain your choice.

Write the time complexity of your preferred algorithm.

Learn Counting Sort Algorithm in LESS THAN 6 MINUTES!