Assignment 1: Complexity Analysis of Algorithms

Overview

Through this assignment the students will learn concepts such as algorithm efficiency and classes of algorithm complexities.

Rules and Deliverables

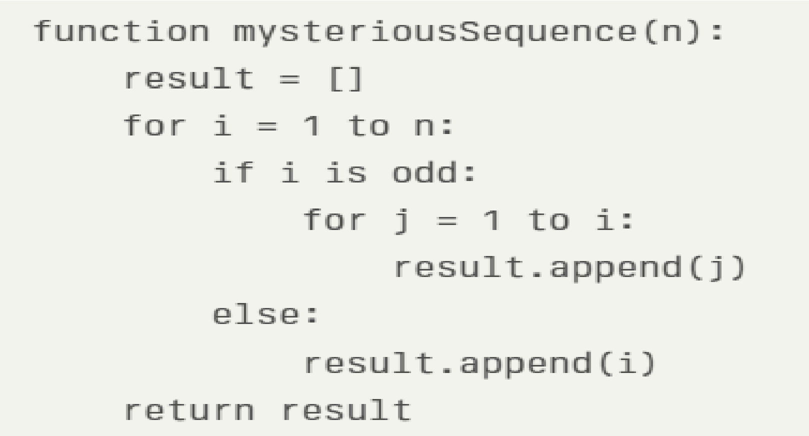
- This is an individual assignment.
- Cheating of any kind is NOT tolerated! Assignments will be checked against each other, and illegal collaboration will be treated based on the University dishonesty policy.
- The due date will be **Saturday 2/8/2025 at 11:59 pm.**
- Submitting the assignment 24 hours after the due date will result in a deduction of 20% from the student's grade.
- Each student should submit the answers document in a PDF format.
- The assignment must be submitted only through Canvas.

Assignment Description

1. Find the Time and Space complexities of the following pseudo-codes.

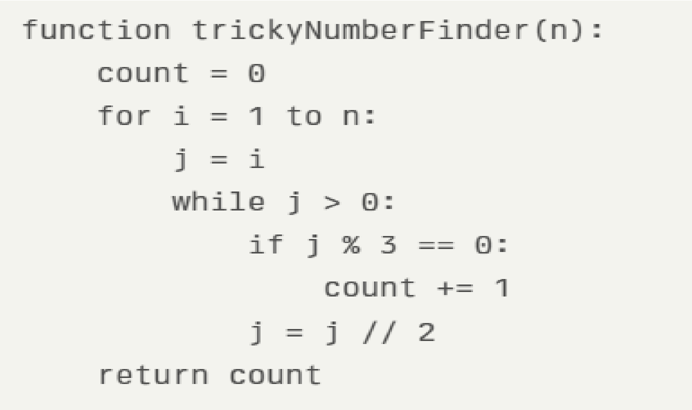
   Assume the space complexity equals auxiliary space + space use by input/output values. **Explain** your answers. (30 points)

   A.



```
function mysteriousSequence(n):
    result = []
    for i = 1 to n:
        if i is odd:
            for j = 1 to i:
                result.append(j)
        else:
            result.append(i)
    return result
```

   B.



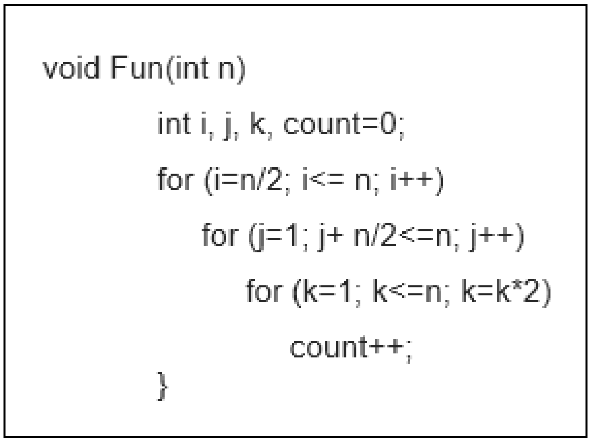
```
function trickyNumberFinder(n):
    count = 0
    for i = 1 to n:
        j = i
        while j > 0:
            if j % 3 == 0:
                count += 1
            j = j // 2
    return count
```

2. Use substitution method and find the close form complexity function for

   T(n) = a T(n/b) + f(n),

   Consider the case where a=1, b=2 and f(n)=1, T(1)=1 (10 points)

3. For the following pseudo-code, what is the function of number of operations (T(n)) and time complexity (O(n))? You can ignore the overhead operations and just count the basic operations. **Explain** your answer. (20 points)



```
void Fun(int n)
        int i, j, k, count=0;
        for (i=n/2; i<= n; i++)
           for (j=1; j+ n/2<=n; j++)
              for (k=1; k<=n; k=k*2)
                 count++;
        }
```