DS2030 DSA for DS
Practise Questions Set 1

1. Give a pseudo-code for an efficient recursive algorithm for reversing a singly linked list.
2. We have an array based stack implementation that increments the array size by 5 elements every time the array becomes full. Starting with an empty stack, what is the cost of 16 pushes?
3. Fill in the blanks.
   (a) The maximum number of nodes in a binary tree of height H is _____
   (b) The maximum number of leaf nodes in a binary tree of height H is _____
   (c) The maximum number of internal nodes in a binary tree of height H is _____
4. A double ended queue or deque is a queue like data structure that supports insertion and deletion at both the front and back of the queue. Write the algorithms the insertion and deletion operations when the deque is implemented using two stacks.
5. Describe in detail an algorithm for reversing a singly linked list L by only redirecting the existing next links (the number of additional nodes created is a constant).
6. Let B be an array of size $n \geq 6$ containing integers from 1 to $n-5$ inclusive, five of which are repeated. Describe an algorithm for finding the five integers in B that are repeated.
7. Indicate for each pair of expressions (A, B), in the table below, whether A is O, Ω of B. Assume that $k \geq 1, c > 1$ are constants. Your answer should be in the form of Y or N written in each box.

A	B	O	Ω
n^2	$nlogn$		
n^k	c^n		
$logn!$	$logn^n$		
n^{logc}	c^{logn}		

8. Give a big-Oh characterisation, in terms of n, of the running time of the `countPrefixSumSums` method shown in the Code 1.

```
def countPrefixSumSums(arr1, arr2):
    # Calculate prefix sums of arr1
    prefix_sums = [0]
    for num in arr1:
        prefix_sums.append(prefix_sums[-1] + num)

    # Initialize count to 0
    count = 0

    # Iterate through arr2
    for num2 in arr2:
        # Check if num2 is in the prefix sums of arr1
        if num2 in prefix_sums:
            count += 1
```

```
15
16      return count
```

Code 1: Sample algorithm for analysis

9. Given a circular linked list, where the last node points back to the first node, forming a cycle, design an algorithm to convert this circular linked list into a linear linked list without any cycles. Describe the algorithm `convertToLinear` that takes the head node of the circular linked list and the element from where linearization should take place as input and returns the head node of the linear linked list. You can assume that the linked list has unique elements. For example, consider the circular linked list: $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 1$. Calling your algorithm with (head and 3) as input should return the linear linked list: $3 \to 4 \to 5 \to 6 \to 1 \to 2$.

10. Show that if $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$, then $f_1(n) \times f_2(n)$ is $O(g_1(n) \times g_2(n))$.

11. A binary tree is a tree in which each node has 0, 1, or 2 children. Given 3 nodes, there are exactly 5 binary trees that can be constructed using these nodes, namely,

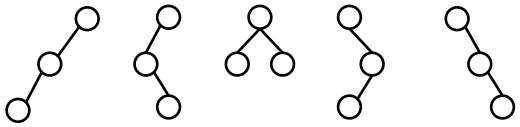

Figure 1: binary trees with three nodes

Write a method `nBTrees(n)` that returns the number of binary trees that can be constructed with n nodes, where $n \geq 0$ What is the complexity of the method?