CS210 Major Exam I (171) with Dr. Basit Qureshi Name__SOLUTION_____ ID_____
Time: 45 minutes Weight: 15 points
Q1 [9]. For each of the following provide an appropriate answer.

/15

(A-2) Give the best Big-O characterization for each of the following running time estimates (where *n* is the size of the input problem).

	Running time estimate	Big-O characterization
(a)	$\log(n) + 10000$	O(log n)
(b)	$2^{10} + 3^5$	O(1)
(c)	$1+2+ \ldots +(n-2)+(n-1)+n$	$O(n^2)$
(d)	$n \log n+15\ n + 0.002\ n^2$	$O(n^2)$

(B-1) Give the time complexity in Big-O notation.

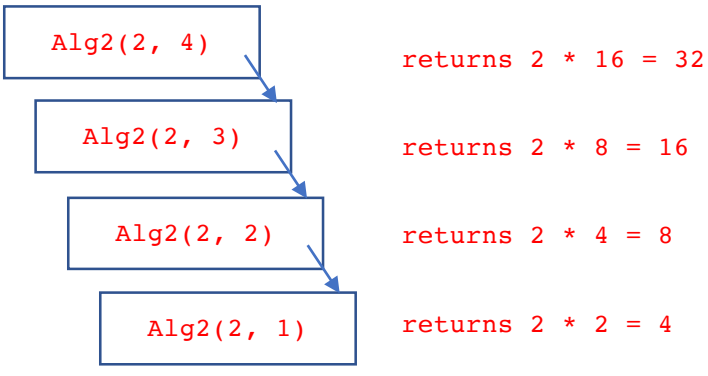
```
public static int Alg1 (int[] groupA, int[] groupB, int[] groupC)
{
   int counter=0;
   for (int a: groupA)
      for (int b: groupB)
        for (int c: groupC)
           if((a==b) && (b==c))
                 counter++;
   return counter;
}
```

Estimate Number of operations: T(n)= Best: $3n^3 + 2$ Worst: $6n^3 + 2$

Time complexity in Big-O notation $O(n^3)$

(C-1) Draw recursion trace (tree) for `Alg2 (2,4)` **and** state its time complexity in Big-O.

```
public static int Alg2(int m, int n)
{
   if ( n == 1 )
       return m;
   else
       return m + Alg2 (m, n-1);
}
```



run time T(n) = cn
Big Oh: O(n) which is linear

(D-1) Given the runtimes of two algorithms with estimated run times T and P, which runs faster? why?

- $T(n) = 3n + \log(n)$
- $P(n) = 4n$

Lets say T(n) runs faster, then we have to satisfy $T(n) \leq P(n)$ for some value n_0.

$3n + \log(n) \leq 4n$
$0 \leq 4n - 3n - \log(n)$
$0 \leq n - \log(n)$
assume $n_0 = 2$ then
$0 \leq 2 - 1$
$0 \leq 1$ satisfied. This shows for $n_0=2$, T(n) runs faster than P(n)

(E-2) Answer each of the following by True [T] or False [F].

T/F

[F]	Removal of a node from circular list is faster compared to a singly list.
[F]	Item insertion/removal from a linked list require data movement (shifting entries for each updating method).
[T]	Doubly-linked lists provide more efficient implementation for ordered data (sorted) than singly linked lists.
[T]	In an array-based implementation of a stack, operations pop() and push() takes constant time.

(F-2) Consider the implementation of a Queue using a single dimension array of maxSize 5. Illustrate the contents of the Queue Q for the following calls.

Call	Return value	Contents of Q
Dequeue()	Error	
Enqueue(5)		5
Enqueue(6)		5 6
Top()	5	5 6
Enqueue(10)		5 6 10
Dequeue()	5	6 10
Enqueue(20)		6 10 20
Enqueue(30)		6 10 20 30
Enqueue(Top() + 10)		6 10 20 30 16
Size()	5	6 10 20 30 16
Enqueue(0)	Error	6 10 20 30 16
Dequeue()	6	10 20 30 16
Dequeue()	10	20 30 16
Enqueue(Size() + 10)		20 30 16 13

Q2.[3] Extend the Singly Linked List class by write a java method `public int frequency(int f)` that counts and returns the number of objects where `Node.id` equals the value in `f`.

```
public int frequency(int f)
{
   Node temp = Head;
   int count=0;
   while(temp!=null)
   {
     if(temp.id==f)
       count++;
     temp = temp.next;
   }
   return count;
}
```

Q3. [3] Write a method `public void reverseOrder(String [] words, Stack S)` that takes two parameters, an array of strings containing words and an empty Stack S of type `String`. This method reverses the order of words given in the String array and prints the contents of the array in reverse order. You must use the provided Stack.

Sample Input: `reverseOrder({“aaa”, “bbb”, “ccc”}, S)`
Sample Output:

```
ccc
bbb
aaa
```

```
public void reverseOrder(String [] words, Stack S)
{

      for(int i=0;i<words.length;i++)
        S.push(words[i]);
      for(int i=0;!S.isEmpty();i++)
      {
         words[i] = S.pop();
         System.out.println(words[i];
      }
}
```

// END OF EXAM