CIS 360 Lab #2: Analyze Algorithms

Name Anubhav Shankar **ID** 01951462

Analyze Algorithms

- Ignore overhead instructions, count only **control instructions**: the number of times these instructions execute **increases** with input size n.
- For loop = running time inside the loop * the number of iterations
- Nested loop = running time inside the nested loops * the product sizes of all the loops

Task A. (Ch1-32) What is the time complexity T(n) of the nested loops below?

Solution: T(n) = O(nlogn). Assume that n is a power of 2: $n = 2^k$ for some positive integer k.

```
for ( i = 1 ; i <= n ; i++) {
        j = n ;
        while (j >= 1) {
                < body o f the while loop> //counted as  ( 1 operation ) .
                j = floor(j/2) ;
        }
}
```

Implement the above procedure, count the number of time the body of the while loop is executed for n =10, 1000, and 100,000

T(10)= 40 T(1000)= 10,000

T(100,000) = 1,700,000

Task B. (Ch1-35) Consider the following algorithm, where the array A is indexed 1 to n:

```
int add_them ( int n , int A[ ] ) {
        index i , j , k ;
        j = 0 ;
        for ( i = 1 ; i <= n ; i++)
                j = j + A[i] ;
        k = 1 ;
        for ( i = 1 ; i <= n ; i++)
                k = k + k ;
        return j + k ;
}
```

(a) **Implement** the above procedure. Execute it with n = 5 and the array A contains 2, 5, 3, 7, and 8, what is returned? Solution: 57

(b) What is the time complexity T(n) of the algorithm?

Solution: T(n) = O(n)

(c) Try to improve the efficiency of the algorithm. Solution:

```
public static int addThem(int n, int[] A) {
  int i,j,k;
  j = 0;
  k=1;

  for(i = 0; i<n; i++) {
   j = j + A[i];
   k = 2*k;
  }
  return j + k;
 }
```