CS211: Algorithms & Data structures

Dr. Sameer M. Alrehaili

September 20, 2021

srehaili@taibahu.edu.sa
college of computer science and engineering ,yanbu, Taibah University

Lab03

1 Laboratory Objectives:

- To practise Time Complexity measuring using primitve operations count method.
- To discuss some examples of worst, average and best case running time

2 Exercises

We will take the problems of sum, maximum numbers, and the addition of two matrices. Time complexity analysis based on primitive operations will be required for each problem.

1. Measure the time complexity of the following Java program using operation count method?

Listing 1: GCD

```
// Hello.java
public static int sum(int[] arr){
   int s=0;
   for(int i=0;i<a.length;i++)
   {
      s+=a[i];
   }
   return s;
}
```

The solution is : $\mathbf{T(n)} = 1 + 1 + (n + 1) + 2n + 3n + 1 = \mathbf{6n+4}$
So, this algorithm runs in $\mathcal{O}(n)$, which is in linear time.

2. What is the time complexity of the following algorithm, use basic operations count?

Algorithm 1: Finding the maximum number of three numbers

```
Input: $a, b$, and $c$, are three numbers
Output: $max$, the maximum number
1: $max \leftarrow a$
2: if $b > max$ then
3:     $max \leftarrow b$
4: end if
5: if $c > max$ then
6:     $max \leftarrow c$
7: end if
8: return $max$
```

The solution is : $\mathbf{T(n)} = 1 + 1 + 1 + 1 + 1 + 1 = \mathbf{6}$
So, this algorithm runs in $\mathcal{O}(1)$, which is in constant time.

3. What is the time complexity of the following Java code, use basic operations count?

Listing 2: GCD

```java
public static int[][] add(int[][] a, int[][] b) {
   int row = a.length;
     int column = a[0].length;
     int[][] c = new int[row][column];

     for (int i = 0; i < row; i++) {
         for (int j = 0; j < column; j++) {
             c[i][j] = a[i][j] + b[i][j];
         }
     }
     return c;
 }
```

The solution is : $\mathbf{T(n)} = 1+2+1+1+(n+1)+2n+n+n^2+n+2n^2+5n^2 = 8n^2 + 5n + 6$
So, this algorithm runs in $\mathcal{O}(n^2)$, which is in quadratic time.