CS211: Algorithms & Data structures

Dr. Sameer M. Alrehaili

September 28, 2021

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

Assignment 2 Solution

1. Calculate the total number of primitive operations executed for the following algorithm?

Algorithm 1: Computing the power of a number

```
   Input:$x$, is a real number $x \in \boldsymbol{R}$, $n$ is an integer number, $n \in \boldsymbol{Z}$
   Output: $x^n$
1: $p \leftarrow 1$
2: for $i \leftarrow 1$ to $n$ do
3:     $p \leftarrow p \times x$
4: end for
5: return $p$
```

$T(n) = cn$, where c is some constant and n is the size of the input. At line (1), we count one unit for initialising p and it executed 1 time, so the total of the first line is $1 \times 1 = 1$. At line (2), we count one unit for initialising i (1×1) at the begining of the for-loop, one unit for testing $i \leq n$ each time we go around the loop $+1$. In addition, two units for incrementing i each time we go around the for-loop $(2n)$. Therefore, the total of the second line is $1 + (n + 1) + 2n = 3n + 2$. At line (3), we count two units for each time we go around the loop $(2n)$. At line(5), we count one unit for returning statement.

$\mathbf{T(n)} = 1 + 3n + 2 + 2n + 1 = \mathbf{5n{+}4}$
Since $5n$ is the highest term in the function we can say $T(n)$ grows at the order of n and we write: $\mathbf{T(n)} = \mathcal{O}(n)$.