Matthew Pelter

Professor Alrajab

CSC 229

21 February 2022

Lab 03 Assignment

1. Find and show how to compute Big O for the following algorithm

public static int getLargest(int arr[], int sz){

int iterate1 = 0; // 1

int iterate2 = 0; // 1

int largest = 0; // 1

while(iterate1 < sz - 1) { // n

iterate2++; // n

if(iterate2 == sz) { // n

iterate1++; // n

iterate2 = iterate1; // n

continue; // n

}

int product = arr[iterate1] \* arr[iterate2]; // n

if(product > largest) // n

largest = product; // n

}

return largest; // 1

}

1 + 1 + 1 + n + n + n + n + n + n + n + n + n + 1

4 + 9n = **O(n)**

1. You have 20 M&Ms bags. 19 bags have 1.0 gram pieces, but one has pieces of weight 1.1 grams. Given a scale that provides an exact measurement, how would you find the heavy bag? You can only use the scale once.

So, if we have 20 bags of M&M’s and one of them has a mass of 1.1grams per m&m, for each bag (n), we place n amount of m&m’s on the scale. So, bag 1, we put 1 m&m, bag 2, 2 m&m, etc. If they were all 1 gram, we would read a mass of 210 grams. Now say bag 3 had the bag with the 1.1-gram mass. We would get a mass of 210.2 grams. If we look at the last 2 digits, we have 02 showing that bag 2 has the 1.1-gram bag.

One more example

Now bag 14 has the 1.1 gram m&m’s. Reading the scale, we see 211.4 grams. Take the last 2 digits and we have 14. Bag 14 is the culprit

The End.