1. Consider the following algorithm (pseudo-code):

int algorithm( A:int[ ] of length n; n > 0 )

x A.length-1

y A[0]

for i 1 to x do

if (A[ i ] > y)

y A[ i ]

return y

(5 points) What does this algorithm do?

Finds the greatest value in an array

(5 points) What is/are the basic operation(s) of this algorithm (5 points) How many basic operations will be performed in total, given an input size of 325 items

Initialize y&i, compare i against length, increment i, compare a[i] to y and adjust values accordingly. Really only 325 because the others are trivial

(5 points) How many basic operations will be performed in total, given an input size of 1000 items?

1000

(5 points) What is the BigO complexity of this algorithm> Explain your analysis, and how you arrived at this conclusion

n because it’s n comparisons of y against each element in the array

(10 points) Write this algorithm in C#, and output total run time, and total operation count. (Please submit your code as a .cs file)

2. Consider the following algorithm:

void func(int[] arr)

{

int n = arr.Length;

for (int i = 1; i < n; ++i) {

int mm = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > mm) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

(5 points) What does this algorithm do?

Sorts ascending?

(5 points) What is/are the basic operation(s) of this algorithm? (cut/copy/paste your code to demonstate)

i < n of arr[j + 1] = arr[j];

i < n of arr[j + 1] = key;

(5 points) How many basic operations will be performed in total, given an input size of 325 items

325

(5 points) What is the BigO complexity of this algorithm. Explain your analysis, and how you arrived at this conclusion

N because there is only one for loop

(10 points) Write this algorithm in C#, and output total run time, and total operation count. (Please submit your code as a .cs file)

3.

(10 points) Write a function in C# that takes a string parameter and returns true if it is a palindrome. (Note. a word, phrase, or sequence that reads the same backward as forward, e.g., “madam” or “nurses run”.) (Please submit your code as a .cs file)

(5 points) What is/are the basic operation(s) of this algorithm

str = str.ToUpper();

string str2 =str;

str2.Reverse();

if (str2.Equals(str))

(5 points) How many basic operations will be performed in total, given an input size of 20 words in the input string?

4

(5 points) What is the BigO complexity of this algorithm. Explain your analysis, and how you arrived at this conclusion

I think it’s 1 because only one string is ever passed in and it doesn’t matter how many words are in the string

4.

(10 points) Write a function that takes an array parameter and only prints the distinct elements (ignoring duplicates.) (i.e. Input = [5,2,7,2,4,7,8,2,3] Output = 5 2 7 4 83) (Please submit your code as a .cs file)

(5 points) What is/are the basic operation(s) of this algorithm (cut/copy/paste your code)

for(int i=0; i<array.Length; i++) if (!temp.Contains(array[i])),

foreach(var item in temp) Console.Write(item+" ");

(5 points) How many basic operations will be performed in total, given an input size of 2000 numbers in the input array?

4000

(5 points) What is the BigO complexity of this algorithm. Explain your analysis, and how you arrived at this conclusion

2n because there are 2 non-nested for loops