ENSF 594 – Principles of Software Development II

Summer 2021

Lab Assignment # 1

Analysis of Algorithm

Total Marks: 25

Due Date: Friday, July 09, by 11:59 PM using D2L

Question 1: (1 marks)

For algorithm A, If the exact number of steps is $T(n)=2n+3n^2-1$ what is the Big O? Explain.

Question 2: (2 marks)

Consider the below functions we discussed in our lecture:

Linear, logarithmic, exponential, quadratic, constant, cubic,

Write the above function from top to bottom order from most to least efficient.

Question 3: (3 marks)

Consider the below code fragment:

```
int test = 0;
   for (int i = 0; i < n; i++){
      for (int j = 0; j < n; j++){
         test = test + i * j;
      }
   }
```

What is its Big-O running time? Explain your answer.

Question 4: (3 marks)

Consider the below code fragment:

```
int func(){
   int test = 0;
   for (int i = 0; i < n; i++){
      test = test + 1;
   }
   for (int j = 0; j < n; j++){
      test = test - 1;
   }
   return 0;
}
```

What is its Big-O running time? Explain your answer.

Question 5: (4 marks)

Consider the below code fragment:

```
int func(){
   int i = n;
   int count = 0;
   while (i > 0){
      count = count + 1;
      i = i // 2;
   }
   return 0;
}
```

What is its Big-O running time? Explain your answer.

Question 6: Write a scenario (or a code fragment), whose complexity is $O(n^3)$ **(3 marks)**

Question 7: If an algorithm performing at $O(n^2)$ has the integer 7 as input, what is the worst case scenario for the algorithm? **(1 marks)**

Question 8: Use Big O Notation to describe the time complexity of the following function that determines whether a given year is a leap year: **(1 marks)**

```
bool isLeapYear(year) {
        return (year % 100 === 0) ? (year % 400 === 0) : (year % 4 === 0);
      }
```

Question 9: Use Big O Notation to describe the time complexity of this function, which is below: **(3 marks)**

```
int chessboardSpace(numberOfGrains)
{   chessboardSpaces = 1;
    placedGrains = 1;
    while (placedGrains < numberOfGrains) {
        placedGrains *= 2;
        chessboardSpaces += 1;
}
return chessboardSpaces; }
```

Explain your answer.

Question 10: Consider the code below: **(4 marks)**

```
i = 1;
sum = 0;
while (i <= n) {
   i = i + 1;
   sum = sum + i;
}
```

In our lecture, we have done an example about calculating the primitive operations and then determines the complexity. First identify the primitive operation of every line, and then calculate the Big-O of the above code? Also mention the class of growth rate function.

Question 11: In our lecture, we have discussed the Big Omega represents the lower bond. What is the lower bound of the below function:

3nlogn – 2n

Notes for Submission:

You should submit a single PDF file for all the questions in this lab assignments. Submit clearly the question number in your pdf file. Use D2L to submit the pdf file