Fundamental of Computer Science

Homework Set 5

November 22, 2023

1. (1') Convert the following pseudocode routine to an equivalent routine using a **repeat** statement. You can add other statements if needed (Hint: consider different values of X).

```
while (X < 8) do (Z ← Z + X;   X ← X + 1)
```

Begin Pseudocode

```
z←0
x←1
repeat (z←z + x;
          x←x+1)
until (x=6)
```

End Pseudocode

2. (1') What sequence of numbers is printed by the following algorithm if it is started with input values 0 and 1? Fill your answer in the form (one line, separated each printed value by one space, e.g., "1 2 3 4 5" without quotes).

```
procedure MysteryWrite (Last, Current)
if (Current < 100) then
   (print the value assigned to Current;
     Temp ← Current + Last;
     apply MysteryWrite to the values Current and Temp)
```

Your Answer:	1 1 2 3 5 8 13 21 34 55 89

3. (2’) Suppose we apply both Test1 and Test2 to the input value 1. Write down the printed outputs of the two routines. Fill your answer in the form (one line, separated each printed value by one space, e.g., “1 2 3 4 5” without quotes).

```
procedure Test1 (Count)
if (Count is not equal to 5)
   then (print the value assigned to Count; apply Test1 to the value Count+1)

procedure Test2 (Count)
if (Count is not equal to 5)
   then (apply Test2 to the value Count+1; print the value assigned to Count)
```

The answer of Test1(1)	1 2 3 4
The answer of Test2(1)	4 3 2 1

4. (3’) The Euclidean algorithm finds the greatest common divisor of two positive integers X and Y by the following process:

> As long as the value of neither X nor Y is zero, continue dividing the larger of the values by the smaller and assigning X and Y the values of the divisor and remainder, respectively. The final value of X is the greatest common divisor.

a) Express the algorithm in pseudocode using an **iterative** structure.

b) Express the algorithm in pseudocode using a **recursive** structure.

Complete the middle part of the algorithm with pseudocode, and don’t modify the existing pseudocode. Pseudocode format: “X mod Y” , “X ← Y” , “X ← Function(A,B)”, “if X==Y then”, “while (X>0) do”.

Begin Pseudocode Euclidean Iterative

```
function GCD_iterative (X, Y)
if X<Y then
     Swap(X,Y)
while (Y>0 ) do
        X ← Y
        Y ← X mod Y
GCD ← X
Return GCD
```

End Pseudocode Euclidean Iterative

Begin Pseudocode Euclidean Recursive

```
function GCD_recursive (X, Y)
if X<Y then
    Swap(X,Y)
if Y==0 then
    GCD ← X
else
     Y ← X mod Y
     GCD_recursive(X,Y)
Return GCD
```

End Pseudocode Euclidean Recursive

5. (1’) List the classes A.Θ(n^2), B.Θ($lg\ n$), C.Θ($n*lg\ n$), D.Θ($\surd n$), E.Θ(n), and F.Θ(n^3) in decreasing order of efficiency. Fill the six uppercase letters into the form, without separators such as spaces (e.g. “ABDCFE” without quotes).

Your Answer	BCDEAF

6. (2’)

a) What is the **maximum** number of entries that must be interrogated when applying the binary search below to a list of 1000 entries? What about a list of 100,000 entries? Fill an integer in each form.

Binary Search Algorithm

(Find the address of an object ‘Target’)

```
function BinarySearch (Array, Target)
    left ← 0
    right ← length(Array) - 1
    while (left <= right)
        middle = RoundDown((left + right) / 2)
        data = the middle-th element of Array
        if data=Target then
            return middle
        else if data<Target then
            left = middle + 1
        else
            right = middle - 1
    return TargetDoesNotExist
```

Question 1:	10
Question 2:	17

b) Suppose we find that a machine programmed with our **insertion sort** algorithm requires an average of one second to sort a list of 100 names. How long do you estimate it takes to sort a list of 1000 names? How about 10,000 names? Please estimate according to the time complexity of **insertion sort**, and fill an integer in each form.

Question 1 (seconds):	100
Question 2 (seconds):	1000000