University of M'sila
Faculty of Mathematics and Computer Science
Department of Computer Science

جامعة المسيلة
كلية الرياضيات و الإعلام الآلي
قسم الإعلام الآلي

Exam : Data Structures and Algorithms 3 (DSA 3)

Duration: 1h 30 - Documents: not allowed **Academic year : 2023 / 2024**

Date : 17/01/2022 (10 : 30 – 12 : 00) **Level : L2 Semester : 5**

Exercise 1 : (1.75 Points / 10 Minutes)

Consider the following function and the snapshot of its local variables in the stack frame (activation record) at **Stack Frame Snapshot**:

```
void main() {
 int i = 3, *p = 0, **r = 0;
 double d = 2.0, e = 0.0, *q = &d;

 p = ……;

 r = ……;

 **r = ……;

 // Stack Frame Snapshot
 printf("Line 1: %p %p %p\n", &i, &d, &e);
 printf("Line 2: %d %lf %p\n", *p, *q, *r);
 printf("Line 3: %p %p %p\n", p, q, r);
 printf("Line 4: %p %p %p\n", &p, &q, &r);
}
```

Stack Frame of main()

at Stack Frame Snapshot

Memory Address	Stack	Stored Variable
0x007AF960	0x007AF96C	p
0x007AF964	0x007AF960	r
0x007AF968	0x007AF970	q
0x007AF96C	4	i
0x007AF970	2.000000	d
0x007AF978	0.000000	e

Q1) Fill up the blank instructions in the program

Q2) Write the output of the function in the blank lines below.

Line 1: …… …… ……

Line 2: …… …… ……

Line 3: …… …… ……

Line 4: …… …… ……

Exercise 2 : (5 Points / 20 Minutes)

Each of the two functions **P1** and **P2** calculates the value of a polynomial $\boldsymbol{p(x) = a_1 + a_1 x + a_2 x^2 + \cdots + a_n x^n}$ for a value of $\boldsymbol{x}$.

The degree $\boldsymbol{n}$ is an integer and the coefficients $\boldsymbol{a_i}$ are stored in an array $\boldsymbol{a}$[] of type float.

```
float P1 (int a[] , int n , float x){
     s = a[0] ;
     for (int i=1 ; i<=n ; i++)
          s+ = a[i]* pow(x,i) ;
     return s ;
}
```

```
float P2 (int a[] , int n , float x){
     s = a[0] ; q = 1;
     for (int i=1 ; i<=n ; i++) {
          q = q * x ;
          s += a[i]*q ;
     }
     return s ;
}
```

Q1) Calculate the number of multiplications $T1(n)$ that function **P1** performs as a function of n and then express its complexity in **Big-O notation**.

……...

……...

Q2) Calculate the number of multiplications $T2(n)$ that function **P2** performs as a function of n and then express its complexity in **Big-O notation**.

……...

……...

Q3) What can we deduce?

……...

……...

Q4) Write a function **P3** that gives the same result in n multiplications.

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

Q5) Show that $\boldsymbol{log}(\boldsymbol{n}!) \in \mathcal{O}(\boldsymbol{n\ log\ n})$.

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

<u>Exercise 3 :</u> (2.75 Points / 15 Minutes)

This is the beginning of a sorting algorithm trace:

10	56	-2	52	-8	41	13
-8	56	-2	52	10	41	13
-8	-2	56	52	10	41	13
-8	-2	10	52	56	41	13

Q1) Which algorithm is it?

……...

……...

Q2) Complete the table with the following steps, until the table is completely sorted.

Q3) Write a function sorting the array $\boldsymbol{Tab}$ of size $\boldsymbol{n}$ using this algorithm.

……...

……...

……...

……...

……...

……...

……...

……...

……...

Q4) What is the complexity of this algorithm as a function of $\boldsymbol{n}$?

……...

……...

<u>Exercise 4 :</u> (5 Points / 20 Minutes)

We want to write a function **`void`** `separation(`**`int`** `Tab[], `**`int`** `n, `**`int`** `x)`, which applied to an array $\boldsymbol{Tab}$ of $\boldsymbol{n}$ integers, moves the elements of $\boldsymbol{Tab}$ such that :

- All elements less than $\boldsymbol{x}$ are placed at the beginning of the array $\boldsymbol{Tab}$
- All elements greater than $\boldsymbol{x}$ are placed at the end of array $\boldsymbol{Tab}$
- All elements equal to $\boldsymbol{x}$ are placed between these two zones.

For example, if $\boldsymbol{Tab} = [\mathbf{0},\mathbf{4},\mathbf{2},\mathbf{3},\mathbf{1},\mathbf{0},\mathbf{2},\mathbf{1},\mathbf{0},\mathbf{3},\mathbf{4},\mathbf{3},\mathbf{1},\mathbf{2},\mathbf{0}]$ and $\boldsymbol{x} = \mathbf{3}$ after calling `separation(Tab,15,3)` we get $\boldsymbol{Tab} = [\mathbf{0},\mathbf{0},\mathbf{2},\mathbf{1},\mathbf{0},\mathbf{2},\mathbf{1},\mathbf{0},\mathbf{2},\mathbf{1},\mathbf{3},\mathbf{3},\mathbf{4},\mathbf{4}]$.

Each element of $\boldsymbol{Tab}$ will be observed **only once** during the execution of the `separation` function.

Q1) Under what conditions will the `separation` function produce **fewer** than **3** different zones in the modified $\boldsymbol{Tab}$ array?

……...

……...

……...

……...

……...

Q2) Complete the code for the `separation` function below (You can use the **`swap`** function without needing to redefine):

```
1. void separation (int Tab[], int n, int x){
2.   int i=0,j=0,k=n-1;
3.   while ( ……){
4.        if (Tab[j]<x){
5.             ……
6.             ……
7.             ……
8.        }
9.        else{    //1
10.            if (Tab[j]>x) {
11.                 ……
12.                 ……
13.            }
14.            else
15.                 ……
16.           } //else 1
17.       }
18. }
```

Q3) What is the property verified by integer ***i*** throughout the execution of the `separation` function?

……...

……...

……...

Q4) Same question for ***j***.

……...

……...

……...

Q5) Same question for ***k***.

……...

……...

……...

Exercise 5 : (5.5 Points / 20 Minutes)

Q1) Suppose a binary tree is **5** layers deep. What is the maximum number of nodes that can fit in this tree before a new layer needs to be added?

……...

……...

……...

Q2) What is the minimum height of a binary tree containing **35** nodes?

……...

Q3) What is the minimum height of a general tree containing **30** nodes?

……...

Q4) Draw all the binary search trees on the keys **1**, **2** and **3**.

……...

……...

……...

……...

……...

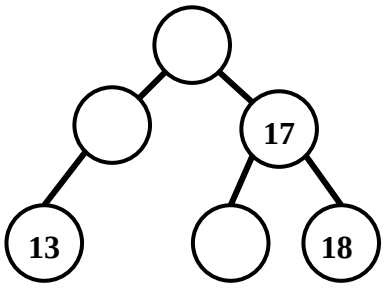
……...

……...

……...

……...

Consider the following (incomplete) binary search tree that stores different integer values :



Q5) Assuming the top node is referenced by root,

what is the value of : `root->left->left->data`

……...

Q6) Fill in the missing integer numbers into the blank nodes.

Q7) Write down the sequence of vales produced by a post-order traversal, assuming values have been filled in as necessary.

……...

……...

Q8) Remove the root, and draw the tree.

……...

……...

……...

……...

……...

……...

……...

……...

A node in a binary tree is defined as follows:

```
struct BTNode {
    int data;
    BTNode *left;
    BTNode *right;
};
typedef BTNode *BTREE;
```

Q9) Write a **Max_BT_It()** function for finding (searching) a maximum element in a binary tree **without recursion (iterative)**.

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

Exercise Bonus : (2 Points / 5 Minutes)

Give the recursive relation $\boldsymbol{T(n)}$ for the running time of the following code snippet and solve its complexity using the recursive substitution method.

```
int bar(int n){
     if (n<=10){
          return 0;
     }
    int r = 0;
     for(int i=0; i<n; i++){
          r++
     }
     return bar(n/3) + bar(2*n/3);
}
```

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

……...

Good Luck