King Saud University	College of Computer and Information Sciences Department of Computer Science
Data Structures CSC 212	**Final Exam - Fall 2019**
Date: 21/12/2019	Duration: 3 hours

Guidelines: No calculators or any other electronic devices are allowed in this exam.

Student ID:	Name:
Section:	Instructor:

1.1	1.2	2	3	4	5	6	7	8	Total

Question 1 .... 14 points

(a) (6 points) Choose the run time from A to D for each of the following cases:

A. $O(1)$. **B**. $O(\log n)$. **C**. $O(n)$. **D**. $O(n \log n)$. **E**. $O(n^2)$.

1. The worst case of `DoubleLinkedList.remove`. ___
2. The worst case of `LinkedPQ.enqueue`. ___
3. The worst case of `BST.insert`. ___
4. The best case of `DoubleLinkedList.remove`. ___
5. The best case of `Heap.insert`. ___
6. The worst case of `ArrayList.findNext`. ___
7. The best case of `LinkedPQ.enqueue`. ___
8. The worst case of `AVL.find`. ___
9. The worst case of `ArrayStack.pop`. ___
10. The worst case of `Heap.insert`. ___
11. The worst case of `Heap sort`. ___
12. The best case of `Heap.remove`. ___

(b) (8 points) Choose the most appropriate data structure for each of the following tasks.

A. LinkedList.	**B**. ArrayList.	**C**. DoubleLinkedList.	**D**. LinkedQueue.
E. LinkedPQueue.	**F**. LinkedStack.	**G**. BT.	**H**. BST.
I. AVL.	**J**. BPlusTree.	**K**. HeapPQueue.	**L**. Graph.

1. An algorithm reads an unknown number of integers from a file, then depending on an input from the user computes one of the following: sum, product, min, max or average of the numbers. ___
2. An algorithm reads the list of flights operated by an airline company in the form $(City_i, City_j)$, meaning there is a flight between the two cities. The algorithm must check whether its is possible to fly from a given city to another using only the flights of this company. ___
3. An algorithm spell-checks the user input by comparing the input text to a set of pre-stored words. ___
4. A hospital wants to manage the waiting list in the emergency service. Cases with the same level of severity are treated according to the order of arrival. ___

Question 2 .... 12 points

(a) (6 points) Fill each entry of the table below by T (for true) or F (for false).

	Time (worst case): $O(n \log n)$	Comparison-based	In-place	Stable
Merge sort				
Quick sort				
Radix sort				

(b) (6 points) Consider the following array where keys are integer and data is of type string: $A = \{(3, B), (5, D), (3, A), (2, E), (7, E), (5, B), (1, F)\}$. We want to sort this array in increasing order. Choose the result produced by the given algorithm from the following options:

Ⓐ $\{(1, F), (2, E), (3, B), (3, A), (5, D), (5, B), (7, E)\}$

Ⓑ $\{(1, F), (2, E), (3, A), (3, B), (5, B), (5, D), (7, E)\}$

Ⓒ $\{(1, F), (2, E), (3, A), (3, B), (5, D), (5, B), (7, E)\}$

Ⓓ $\{(1, F), (2, E), (3, B), (3, A), (5, B), (5, D), (7, E)\}$

Ⓔ None of the above.

1. Heap sort (in sift-down, swap left if children are equal): Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ
2. Bubble sort: Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ

Question 3 .... 12 points

(a) Write the method **private boolean** f(BTNode<T> t, T e, **int** k), member of BT, which return true if e appears in t at a depth that is equal or greater than k (assume that t is at depth 0).

```
1 private boolean f(BTNode<T> t, T e, int k) {
2   if (...)
3     ...;
4   if (...)
5     ...;
6   return ...;
7 }
```

- Line 2:

  Ⓐ `if (e.equals(t.data))`

  Ⓑ `if (t==null)`

  Ⓒ `if (t.left==null && t.right==null)`

  Ⓓ `if (k > 0)`

  Ⓔ None

- Line 3:

  Ⓐ `return true;`

  Ⓑ `return k;`

  Ⓒ `return false;`

  Ⓓ `return k == e.data;`

  Ⓔ None

- Line 4:

  Ⓐ `if (k>0 || e.equald(t.data))`

  Ⓑ `if (k<=0 && e.equald(t.data))`

  Ⓒ `if (k>0 && e.equald(t.data))`

  Ⓓ `if (k<0 && e.equald(t.data))`

  Ⓔ None

- Line 5:

  (A) `return k>0;`

  (B) `return false;`

  (C) `return e.equald(t.data);`

  (D) `return true;`

  (E) None

- Line 6:

  (A) `return f(t.left,e,k-1)&&f(t.right,e,k-1);`

  (B) `return f(t.left,e,k)&&f(t.right,e,k);`

  (C) `return f(t.left,e,k)||f(t.right,e,k);`

  (D) `return f(t.left,e,k-1)||f(t.right,e,k-1);`

  (E) None

(b) Repeat the same questions as above, but this time as a user.

```
1  public static <T> boolean f(BT<T> bt, T e, int k) {
2    if (...)
3      return ...;
4    ...;
5    return ...;
6  }
7  private static <T> int rf(BT<T> bt, T e, int k) {
8    if (...)
9      return ...;
10   if (...) {
11     if (...)
12       return ...;
13     ...;
14   }
15   if (...) {
16     if (...)
17       return ...;
18     ...;
19   }
20   return ...;
21 }
```

- Line 2:

  (A) `if (bt.empty())`

  (B) `if (k == 0)`

  (C) `if (bt.full())`

  (D) `if (!bt.full())`

  (E) None

- Line 3:

  (A) `return false;`

  (B) `return true;`

  (C) `return e.equals(bt.retrieve());`

  (D) `return k == 0;`

  (E) None

- Line 4:

  (A) `bt.find(relative.LeftChild);`

  (B) `bt.find(relative.Parent);`

  (C) `bt.find(relative.RightChild);`

  (D) `bt.find(relative.Parent);`

  (E) None

- Line 5:

  (A) `return rf(bt,e,k);`

  (B) `return rf(bt.left,e,k)||rf(bt.right,e,k);`

  (C) `return rf(bt,e,k-1);`

  (D) `return rf(bt,e,k+1);`

  (E) None

- Line 8:

  (A) `if (k<=0)`

  (B) `if (k<0 && e.equals(bt.retrieve()))`

  (C) `if (k<=0 && e.equals(bt.retrieve()))`

  (D) `if (k==0 || e.equals(bt.retrieve()))`

  (E) None

- Line 9:

  (A) `return true;`

Ⓑ `return k < 0;`

Ⓒ `return e.equals(bt.retrieve());`

Ⓓ `return false;`

Ⓔ None

- Line 10:

  Ⓐ `if (bt.find(Relative.Parent)){`

  Ⓑ `if (bt.find(Relative.Root)){`

  Ⓒ `if (bt.find(Relative.LeftChild)){`

  Ⓓ `if (bt.find(Relative.LeftChild)!=nulll){`

  Ⓔ None

- Line 11:

  Ⓐ `if (rf(bt,e,k-1))`

  Ⓑ `if (rf(bt,e,k))`

  Ⓒ `if (rf(bt,e,k+1))`

  Ⓓ `if (rf(bt,e,2*k))`

  Ⓔ None

- Line 12:

  Ⓐ `return k>0;`

  Ⓑ `return true;`

  Ⓒ `return k<=0;`

  Ⓓ `return false;`

  Ⓔ None

- Line 13:

  Ⓐ `bt.find(Relative.Root);`

  Ⓑ `bt.find(Relative.Parent);`

  Ⓒ `bt.find(Relative.LeftChild);`

  Ⓓ `bt.find(Relative.RightChild);`

  Ⓔ None

- Line 15:

  Ⓐ `if (bt.find(Relative.RightChild)){`

  Ⓑ `if (bt.find(Relative.Root)){`

  Ⓒ `if (bt.find(Relative.RightChild)!=nulll){`

  Ⓓ `if (bt.find(Relative.Parent)){`

  Ⓔ None

- Line 16:

  Ⓐ `if (rf(bt,e,2*k+1))`

  Ⓑ `if (rf(bt,e,k+1))`

  Ⓒ `if (rf(bt,e,k-1))`

  Ⓓ `if (rf(bt,e,k))`

  Ⓔ None

- Line 17:

  Ⓐ `return k<=0;`

  Ⓑ `return false;`

  Ⓒ `return true;`

  Ⓓ `return k>0;`

  Ⓔ None

- Line 18:

  Ⓐ `bt.find(Relative.RightChild);`

  Ⓑ `bt.find(Relative.LeftChild);`

  Ⓒ `bt.find(Relative.Parent);`

  Ⓓ `bt.find(Relative.Root);`

  Ⓔ None

- Line 20:

  Ⓐ `return k<=0;`

  Ⓑ `return false;`

  Ⓒ `return k>=0;`

  Ⓓ `return k==0;`

  Ⓔ None

Question 4 .... 14 points

(a) (4 points) Choose the most appropriate answer.

1. In a min heap:

   Ⓐ All keys at level k are smaller than all keys in level $k+1$. Ⓑ The largest key is always at the last level. Ⓒ The key of the left child of any node is smaller than the key of its right child. Ⓓ All of the above. Ⓔ None of the above.

2. The worst case run time for bottom-up heap construction is:

   Ⓐ $O(1)$. Ⓑ $O(\log n)$. Ⓒ $O(n)$. Ⓓ $O(n \log n)$ Ⓔ $O(n^2)$.

(b) (10 points) Consider the following heap represented as an array: 20, 18, 12, 10, 11, 5, 2, 6, 4, 3. Choose the correct answer for every operation (all operations are done on the above heap).

1. Heap after inserting 19:

   Ⓐ 20, 19, 12, 10, 18, 5, 2, 6, 4, 3, 11 Ⓑ 20, 19, 12, 11, 18, 5, 2, 6 ,4, 11, 3 Ⓒ 20, 18, 12, 10, 11, 5, 2, 6, 4, 3, 19 Ⓓ 20, 18, 12, 10, 19, 5, 2, 6, 4, 3,11 Ⓔ None

2. Heap after inserting 1 **then** 23: Ⓐ 20,18, 23, 10, 11, 12, 2, 6, 4, 3, 1, 5 Ⓑ 20, 18, 12, 10, 11, 5, 2, 6, 4, 3, 1, 23 Ⓒ 23, 18, 20, 10, 11, 12, 2, 6, 4, 3, 1, 5 Ⓓ 23, 20, 18, 10, 11, 12, 2, 6, 4, 3, 1, 5 Ⓔ None

3. Heap after deleting one key: Ⓐ 20, 18, 12, 10, 11, 5, 2, 6, 4 Ⓑ 3, 18, 12, 10, 11, 5, 2, 6, 4 Ⓒ 12, 18, 5, 10, 11, 3, 2, 6, 4 Ⓓ 18, 11, 12, 10, 3, 5, 2, 6, 4 Ⓔ None

4. Heap after deleting two keys: Ⓐ 4, 18, 12, 10, 11, 5, 2, 6 Ⓑ 18, 11, 12, 10, 4, 5, 2, 6 Ⓒ 12, 4, 5, 10, 11, 3, 2, 6 Ⓓ 12, 11, 4, 10, 3, 5, 2, 6 Ⓔ None

Question 5 .... 14 points

(a) (4 points)

Remark 1. *In what follows the depth of tree is the number of levels in the tree from the root to a leaf (counting number of nodes not edges). Hence, tree with 1 node has depth 0.*

Choose the most appropriate answer:

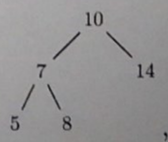
1. The maximum depth of an AVL tree with 8 nodes is:

   Ⓐ 1. Ⓑ 2. Ⓒ 3. Ⓓ 4. Ⓔ 5.

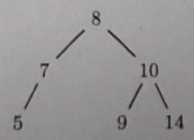
2. The minimum number of rotations caused by an insert in an AVL tree with n nodes and height h is (a single rotation is counted 1; a double rotation is counted 2):

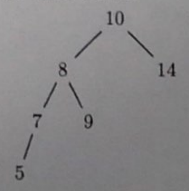
   Ⓐ 0. Ⓑ 1. Ⓒ 2. Ⓓ h Ⓔ n.

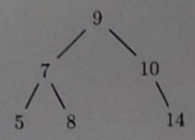
(b) (10 points) Choose the correct result in each of the following cases (follow the the convention of replacing with the smallest key in the right sub-tree when necessary):

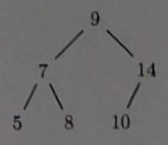
1. After inserting the key 9 in the AVL 

, the tree becomes:

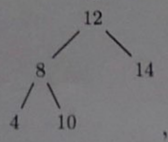
Ⓐ 

Ⓑ 

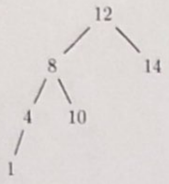
Ⓒ 

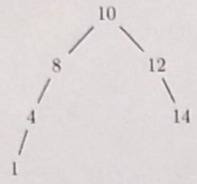
Ⓓ 

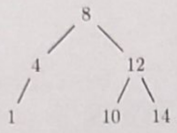
Ⓔ None

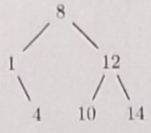
2. After inserting the key 1 in the AVL 

, the tree becomes:

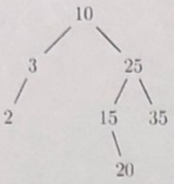
Ⓐ 

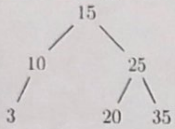
Ⓑ 

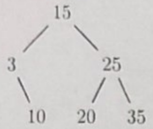
Ⓒ 

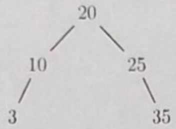
Ⓓ 

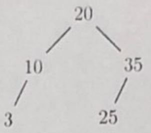
Ⓔ None

3. After deleting the key 2 from the AVL  , the tree becomes:

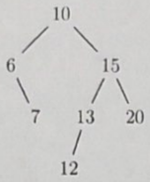
Ⓐ 

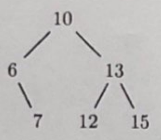
Ⓑ 

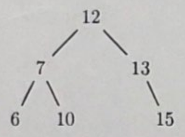
Ⓒ 

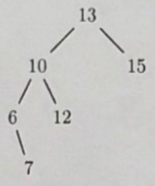
Ⓓ 

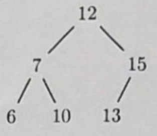
Ⓔ None

4. After deleting the key 20 from the AVL  , the tree becomes:

Ⓐ 

Ⓑ 

Ⓒ 

Ⓓ 

Ⓔ None

Question 6 .... 14 points

(a) (4 points) Choose the most appropriate answer:

1. The insert operation in a hash table using external chaining has a best case run time:

   Ⓐ $O(1)$ Ⓑ $O(\log n)$ Ⓒ $O(n)$ Ⓓ $O(n \log n)$ Ⓔ None

2. How many keys can a hash table that uses folding on a single digit store if the key contains 3 digits?

   Ⓐ 4 Ⓑ 8 Ⓒ 26 Ⓓ 27 Ⓔ 28

(b) (10 points) Use the hash function $H(key) = key\%5$ to store the sequence of keys $25, 13, 14, 23, 16$ in a hash table of size 5. Use the following collision resolution strategies:

1. Linear rehashing (c=1):

Key	25	13	14	23	16
Position					
Number of probes					

2. External chaining:

Key	25	13	14	23	16
Index of the list					

3. Coalesced chaining with cellar size 2 and address region size 5 (put -1 if there is no next element):

Key	25	13	14	23	16
Position					
Index of next element					

Question 7 .... 14 points

(a) (4 points)

Remark 2. *In what follows, a tree with 1 node has depth 1 and height 1. Recall also that a B+ tree has two parameters, m: the maximum number of children and l: the maximum number of elements in a leaf node.*

Choose the most appropriate answer:

1. The minimum number of children of a root node in a B+ tree is:

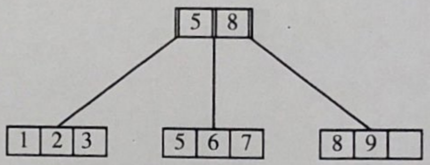
   Ⓐ 0 Ⓑ 1 Ⓒ 2 Ⓓ $\lceil \frac{m}{2} \rceil$ Ⓔ None

2. The maximum number of data elements in a B+ tree with $m = l$ and height h is :

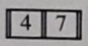
   Ⓐ m^h Ⓑ 2^h Ⓒ h^m Ⓓ $h + m$ Ⓔ None

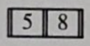
(b) (10 points) Choose the correct result in each of the following cases (when possible, always borrow and transfer to the left):

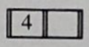
1. After inserting the key 4 in the B+ tree

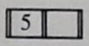


, the **root** becomes:

Ⓐ 

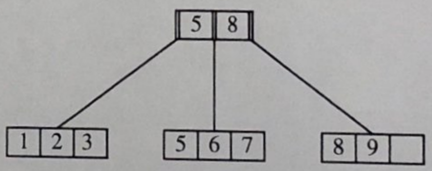
Ⓑ 

Ⓒ 

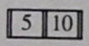
Ⓓ 

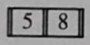
Ⓔ None

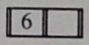
2. After inserting the key 10 in the B+ tree

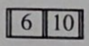


, the **root** becomes:

Ⓐ 

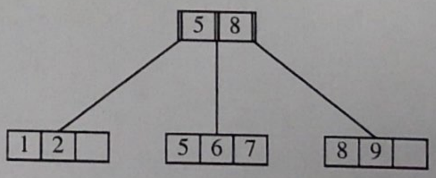
Ⓑ 

Ⓒ 

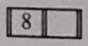
Ⓓ 

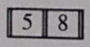
Ⓔ None

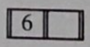
3. After deleting the key 2 from the B+ tree

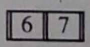


, the **root** becomes:

Ⓐ 

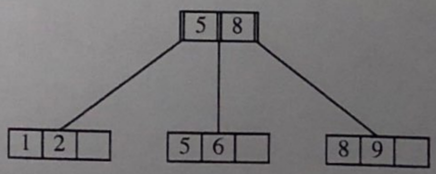
Ⓑ 

Ⓒ 

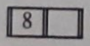
Ⓓ 

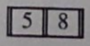
Ⓔ None

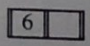
4. After deleting the key 5 from the B+ tree

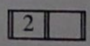


, the **root** becomes:

Ⓐ 

Ⓑ 

Ⓒ 

Ⓓ 

Ⓔ None

Question 8 .... 6 points

(a) (2 points) Choose the most appropriate answer:

1. Adjacency matrix takes $O(n^2)$ memory space, when is it appropriate to use it instead of adjacency list:

   Ⓐ The graph is sparse. Ⓑ The graph is dense. Ⓒ The graph is unweighted Ⓓ The number of edges is less than the number of nodes. Ⓔ None.

2. You apply DFS from a given node and you find out that all nodes were visited. This means:

   Ⓐ The graph is directed. Ⓑ Some nodes have no edges. Ⓒ The graph is connected. Ⓓ The graph contains cycles. Ⓔ None.

(b) (4 points) Given the following graph adjacency matrix, answer the questions below.

	A	B	C	D	E	F
A		3	3			5
B	3		1	5		
C	3	1				
D		5				2
E						
F	5			2		

1. Which of the following sequences are simple paths in this graph? Answer by T (true) or F (false).

   (a) $(C, B, D, E,)$ ___

   (b) (C, B, A) ___

   (c) (C, A, F, D, B) ___

   (d) (B, A, F, D, B, C) ___

2. Answer by T (true) or F (false).

   (a) The graph is connected. ___

   (b) The number of edges in the graph is 6. ___

   (c) (A, B, D, F, A) is a cycle. ___

   (d) The shortest path from F to B is (F, A, B). ___

3. The BFS traversal of this graph starting from A is (insert neighbors in the data structure in increasing alphabetic order):

   Ⓐ A, D, B, C, F. Ⓑ A, B, C, D, F. Ⓒ A, B, C, F, D. Ⓓ A, F, B, C, D. Ⓔ A, C, B, F, D.

4. The DFS traversal of this graph starting from A is (insert neighbors in the data structure in increasing alphabetic order):

   Ⓐ A, F, D, C, B. Ⓑ A, F, C, D, B. Ⓒ A, B, C, F, D. Ⓓ A, F, B, C, D. Ⓔ A, C, B, F, D.