King Saud University	College of Computer and Information Sciences Department of Computer Science
Data Structures CSC 212	**Final Exam - Spring 2021**
Date: 24/04/2021	Duration: 3 hours

Question 1 . . . . 12 points

Choose the most appropriate data structure for each of the following tasks.

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

1. An electronic book reader (e-reader) that allows users to easily flip through pages. C
2. An online store that displays items in a least-expensive to most-expensive order. I
3. A text processing program that prints out strings in a reversed order. C
4. A database management system that stores high-demand magazine articles. [illegible]
5. A social network platform that verifies whether a given username is available or taken. E
6. A food delivery application that helps its drivers deliver multiple orders in the shortest time possible. L

Question 2 . . . . 12 points

Write the method `greaterThanK` which receives a list of integers l and a number k. The method should return a new list containing all the integers in l that are greater than k. If there is no integer in l higher than the value k, then the method should display an appropriate message and returns an empty list.

Example 1. *if List l: [1 2 3 4 5 6 7 8 9] and $k = 5$.*

Then, the method should return the list: [6 7 8 9]

```
public static LinkedList<Integer> greaterThanK(LinkedList<Integer> l, int k) {
  LinkedList<Integer> nl = new LinkedList<Integer>();
  if (.B.)
    System.out.println("List␣is␣empty");
  else {
    .D.
    while (.D.){
      if(.B.)
        nl.insert(.D.);
      B..
    }
    if(B..)
      nl.insert(C.);
    if (.D.)
      System.out.println("no␣value␣in␣L␣is␣greater␣than␣K");
  }
  return B.;
}
```

1. Line 3:

   (A) `l.hasNext()`

   [✓ (B) `l.empty()`]

   (C) `l.full()`

   (D) `l.last()`

   (E) None

2. Line 6:

   (A) `nl.findNext();`

   (B) `nl.findFirst();`

   (C) `l.findNext();`

   [✓ (D) `l.findFirst();`]

   (E) None

3. Line 7:

   (A) `l.retrieve()>= k`

   (B) `l.retrieve()!= k`

   (C) `l.last()`

   [✓ (D) `!l.last()`]

   (E) None

4. Line 8:

   (A) `l.retrieve()>= k`

   [✓ (B) `l.retrieve()>k`]

   (C) `nl.retrieve();`

   (D) `l.retrieve();`

   (E) None

5. Line 9:

   (A) `nl.retrieve()`

   (B) `l.retrieve()!= null`

   (C) `k`

   [✓ (D) `l.retrieve()`]

   (E) None

6. Line 10:

   (A) `nl.findNext();`

   [✓ (B) `l.findNext();`]

   (C) `nl.remove();`

   (D) `l.remove();`

   (E) None

7. Line 12:

   (A) `l.retrieve()>= k`

   [✓ (B) `l.retrieve()>k`]

   (C) `nl.retrieve();`

   (D) `l.retrieve();`

   (E) None

8. Line 13:

   (A) `nl.retrieve()`

   (B) `l.retrieve()!= null`

   [✓ (C) `l.retrieve()`]

   (D) `k`

   (E) None

9. Line 14:

   (A) `l.empty()`

   (B) `l.full()`

   (C) `nl.full()`

   [✓ (D) `nl.empty()`]

   (E) None

10. Line 17:

    (A) `l;`

    [✓ (B) `nl;`]

    (C) `l.retrieve();`

    (D) `n`

    (E) None

Question 3 . . . . 14 points

(a) Write the method `public void trim(int l)`, member of the class `BT`, which removes all nodes that are at level `l` or deeper. We follow the convention that the root is at level 0. Assume that `l` > 0.

```
1  public void trim(int l) {
2    .B
3  }
4  private ... recTrim(...) {
5    if (...)
6      ...
7    if (...) {
8      ...
9      ...
10   } else {
11     ...
12     ...
13   }
14 }
```

1. Line 2:
   - (A) recTrim(root);
   - [✓ (B) recTrim(root, l);]
   - (C) recTrim(null);
   - (D) recTrim(root, root);
   - (E) None ~~→ recTrim([illegible])~~

2. Line 4:
   - [✓ (A) private void recTrim(BTNode<T> t, int l){]
   - (B) private boolean recTrim(BTNode<T> t){
   - (C) private void recTrim(BTNode<T> t){
   - (D) private void recTrim(BTNode<T> t1, BTNode<T> t2){
   - (E) None

3. Line 5:
   - (A) if (t.data == l)
   - [✓ (B) if (t == null)]
   - (C) if (l == null)
   - (D) if (t1 == null || t2 == null)
   - (E) None

4. Line 6:
   - (A) return t1.data.equals(l)|| t2.data.equals(l);
   - (B) return true;
   - [✓ (C) return;]
   - (D) recTrim(t);
   - (E) None

5. Line 7:
   - (A) if (l == 1){
   - (B) if (t1.data == l || t2.data == l){
   - (C) if (t.data == l){
   - [✓ (D) if (l == 0){]
   - (E) None

6. Line 8:
   - (A) t.data = null;
   - (B) t1.left = null;
   - (C) t.left = t.right;
   - [✓ (D) t.left = null;]
   - (E) None

7. Line 9:
   - (A) t2.right = null;
   - (B) t.data = null;
   - (C) t.right = t.left;
   - [✓ (D) t.right = null;]
   - (E) None

8. Line 11:
   - (A) recTrim(t.left);
   - (B) recTrim(t);
   - (C) recTrim(t1.left, t1.right);
   - [✓ (D) recTrim(t.left, l - 1);]
   - (E) None

9. Line 12:
   - [✓ (A) recTrim(t.right, l - 1);]
   - (B) recTrim(t.right);
   - (C) recTrim(t2.left, t2.right);

Ⓓ `recTrim(t);`

Ⓔ None

(b) Write the method **public static** <T> **void** `setNull(BT<T> bt, int l)`, user of the class `BT`, which sets to null the data of all nodes that are at level `l` or deeper. We follow the convention that the root is at level 0.

```
1  public static <T> void setNull(BT<T> bt, int l) {
2    if (...)
3      ...
4    ...
5    ...
6  }
7  private static <T> ... recSetNull(...) {
8    if (...)
9      ...
10   if (...) {
11     ...
12     ...
13   }
14   if (...) {
15     ...
16     ...
17   }
18 }
```

return bt.empty()

1. Line 2:
   - Ⓐ `if (bt.retrieve()== l)`
   - Ⓑ `if (bt.full())`
   - Ⓒ `if (bt.find(l))`
   - [✓ Ⓓ `if (bt.empty())`]
   - Ⓔ None
2. Line 3:
   - Ⓐ `return bt.level();`
   - Ⓑ `bt.update(l);`
   - [✓ Ⓒ `return;`]
   - Ⓓ `return bt.retrieve()== l;`
   - Ⓔ None
3. Line 4:
   - Ⓐ `recSetNull(bt);`
   - Ⓑ `recSetNull(bt, l);`
   - Ⓒ `bt.find(Relative.Parent);`
   - Ⓓ `bt.find(Relative.LeftChild);`
   - Ⓔ None
4. Line 5:
   - Ⓐ `bt.find(Relative.RightChild);`
   - Ⓑ `recSetNull(bt, l);`
   - Ⓒ `bt.find(l);`
   - Ⓓ `bt.find(Relative.Root);`
   - Ⓔ None
5. Line 7:
   - Ⓐ `private static <T> void recSetNull(BT<T> bt){`
   - Ⓑ `private static <T> void recSetNull(BT<T> bt, int l){`
   - Ⓒ `private static <T> boolean recSetNull(BT<T> bt){`
   - Ⓓ `private static <T> void recSetNull(int l){`
   - Ⓔ None
6. Line 8:
   - Ⓐ `if (l <= 0)`
   - Ⓑ `if (bt == null)`
   - Ⓒ `if (bt.level()== l)`
   - [✓ Ⓓ `if (l > 0)`]
   - Ⓔ None

   bt_empty
7. Line 9:
   - Ⓐ `bt.remove();`
   - [✓ Ⓑ `bt.update(null);`]
   - Ⓒ `bt.update(l);`
   - Ⓓ `bt.insert(null);`

Ⓔ None

8. Line 10:

   Ⓐ `if (bt.find(Relative.Parent)){`

   Ⓑ `if (bt.retrieve()== 1)){`

   Ⓒ `if (bt.find(Relative.Root)){`

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

   Ⓔ None

9. Line 11:

   Ⓐ `recSetNull(bt);`

   Ⓑ `recSetNull(l - 1);`

   Ⓒ `bt.find(Relative.Parent);`

   Ⓓ `recSetNull(bt, l - 1);`

   Ⓔ None

10. Line 12:

   Ⓐ `recSetNull(l);`

   Ⓑ `recSetNull(bt);`

   Ⓒ `bt.find(Relative.Root);`

   Ⓓ `bt.find(Relative.Parent);`

   Ⓔ None

11. Line 14:

   Ⓐ `if (bt.find(Relative.Parent)){`

   Ⓑ `if (bt.retrieve()== 1)){`

   Ⓒ `if (bt.find(Relative.Root)){`

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

   Ⓔ None

12. Line 15:

   Ⓐ `bt.find(Relative.Parent);`

   Ⓑ `recSetNull(bt, l - 1);`

   Ⓒ `recSetNull(bt);`

   Ⓓ `recSetNull(l - 1);`

   Ⓔ None

13. Line 16:

   Ⓐ `bt.find(Relative.Parent);`

   Ⓑ `bt.find(Relative.Root);`

   Ⓒ `recSetNull(l);`

   Ⓓ `recSetNull(bt);`

   Ⓔ None

Question 4 .... 14 points

Choose the most appropriate answer.

1. Consider an array A representing a min-heap with 1024 nodes, where the root is stored at position 1. Which of the following is **necessarily** true?

   Ⓐ $A[3] \leq A[1024]$. Ⓑ $A[3] \geq A[1024]$. Ⓒ $A[3] \neq A[1024]$. Ⓓ $A[3] == A[1024]$. Ⓔ None.

2. Suppose T is a binary tree with 128 nodes and height 9 (an empty tree has height 0). Which of the following is **necessarily** false?

   Ⓐ T is a complete tree. Ⓑ T is a min-heap. Ⓒ T is a max-heap. Ⓓ A, B, and C. Ⓔ None.

3. Which traversal algorithm visits the nodes of a max-heap in decreasing order?

   Ⓐ Pre-order Ⓑ In-order Ⓒ Post-order Ⓓ BFS Ⓔ DFS Ⓕ None

4. You have a min-heap of size 63 with the key 1 at root. You insert another key 1, and sift-up stops immediately without doing any swaps. After this insert, the number of keys equal to 1 in the heap is:

   Ⓐ 2. Ⓑ At least 7. Ⓒ At most 7. Ⓓ At most 6. Ⓔ None

5. Given the heap $H = [2, 9, 8, 13, 11, 17, 14]$, what is H after inserting 3 and then 7?

   Ⓐ 2, 3, 8, 7, 13, 9, 11, 14, 17 Ⓑ 2, 3, 8, 7, 11, 17, 14, 9, 13 Ⓒ 2, 3, 7, 8, 9, 11, 13, 14, 17

(D) 2, 3, 7, 9, 11, 8, 14, 17, 13 (E) None

6. Given the heap $H = [1, 5, 2, 8, 7, 16, 10]$, what is H after deleting one key?

(A) 2, 10, 8, 5, 7, 16 (B) 2, 5, 7, 8, 10, 16 (C) 2, 5, 8, 7, 16, 10 (D) 2, 5, 10, 8, 7, 16 (E) None

7. Given the heap $H = [2, 4, 7, 9, 17, 10, 15, 11]$, what is H after deleting two keys?

(A) 2, 4, 7, 9, 17, 10. (B) 7, 9, 10, 11, 17, 15. (C) 7, 9, 17, 10, 15, 11. (D) 7, 9, 10, 11, 15, 17. (E) None

Question 5 .... 14 points

Choose the most appropriate answer:

Remark 1. *Follow the the conventions:*

- *The height of an empty tree is 0.*
- *When necessary, replace with the smallest key in the right sub-tree.*

1. The maximum height of an AVL tree with 7 nodes is:

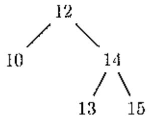
(A) 1. (B) 2. (C) 3. (D) 4. (E) 5.

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

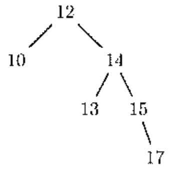
(A) 0. (B) 1. (C) 2. (D) h (E) n.

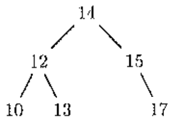
3. The worst case run time for delete in an AVL tree is:

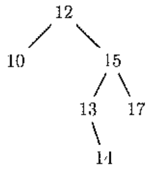
(A) $O(1)$. (B) $O(n)$. (C) $O(\log n)$. (D) $O(n \log n)$ (E) $O(n^2)$.

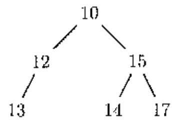
4. After inserting the key 17 in the AVL 

, the tree becomes:

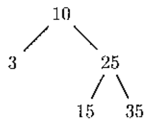
(A) 

(B) 

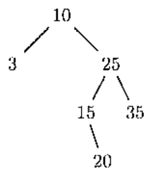
(C) 

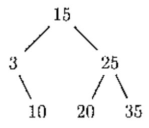
(D) 

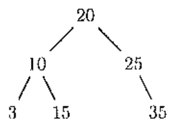
(E) None

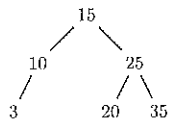
5. After inserting the key 20 in the AVL 

, the tree becomes:

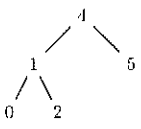
(A) 

(B) 

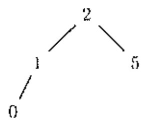
(C) 

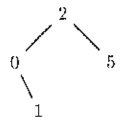
(D) 

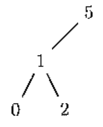
(E) None

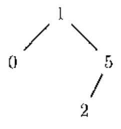
6. After deleting the key 4 from the AVL 

, the tree becomes:

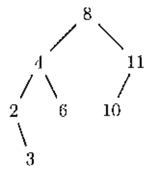
(A) 

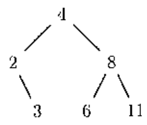
(B) 

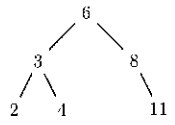
(C) 

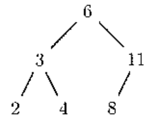
(D) 

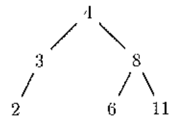
(E) None

7. After deleting the key 10 from the AVL  , the tree becomes:

(A) 

(B) 

(C) 

(D) 

(E) None

Question 6 .... 14 points

Choose the most appropriate answer:

1. What is a hash function?

   (A) A function that allocates memory to keys (B) A function that computes the location of the key in the array (C) A function that creates an array (D) A function that computes the location of the values in the array (E) None

2. A hash table of size 10 uses open addressing with hash function h(k)=k mod 10, and linear re-hashing with $c = 1$. The figure below shows the table after inserting 6 keys. Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

   (A) 46, 42, 34, 52, 23, 33 (B) 34, 42, 23, 52, 33, 46 (C) 46, 34, 42, 23, 52, 33 (D) 42, 46, 33, 23, 34, 52 (E) None of the above.

3. Given the following keys (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function key mod 10, which of the following statements are true?

   i. 9679, 1989, 4199 hash to the same value. ii. 1471, 6171 hash to the same value. iii. All keys hash to the same value. iv. Each key hashes to a different value.

   (A) i only (B) ii only (C) i and ii only (D) iii or iv (E) None of the above.

4. What is the worst case search time of a hashing using separate chaining?

   (A) $O(1)$ (B) $O(\log n)$ (C) $O(n)$ (D) $O(n \log n)$ (E) None of the above.

5. Consider the following hash function: select the two rightmost digits then apply mod 7 on the corresponding number. Which of the following couples of keys cause a collision?

   Ⓐ 3848 and 4756 Ⓑ 3973 and 1258 Ⓒ 162 and 35476 Ⓓ All of the above. Ⓔ None of the above.

Question 7 .... 14 points

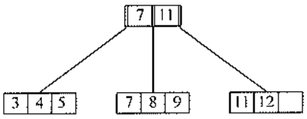
Choose the most appropriate answer (in questions 2 to 7 below, $m = l = 3$):

1. The leaves of a B+ tree of order 4 can contain the following number of elements (suppose $l = m$):

   Ⓐ 3 to 4 elements Ⓑ 2 to 4 elements Ⓒ 1 to 4 elements Ⓓ 4 to 4 elements Ⓔ None

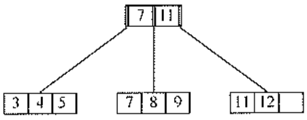
2. After inserting the key 7 in the B+ tree $\boxed{1}\boxed{3}\boxed{6}$, the **root** of tree becomes:

   Ⓐ $\boxed{5}\boxed{}$ Ⓑ $\boxed{5}\boxed{6}$ Ⓒ $\boxed{6}\boxed{}$ Ⓓ $\boxed{7}\boxed{}$ Ⓔ None



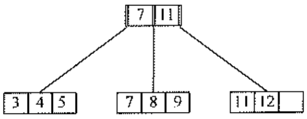
3. After inserting the key 14 in the B+ tree , the **root** of the tree becomes:

   Ⓐ $\boxed{7}\boxed{11}$ Ⓑ $\boxed{7}\boxed{10}$ Ⓒ $\boxed{6}\boxed{}$ Ⓓ $\boxed{8}\boxed{}$ Ⓔ None



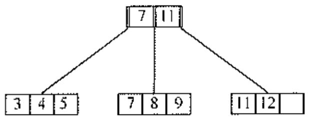
4. After inserting the key 10 in the B+ tree , the **root** of the tree becomes:

   Ⓐ $\boxed{5}\boxed{10}$ Ⓑ $\boxed{7}\boxed{11}$ Ⓒ $\boxed{7}\boxed{10}$ Ⓓ $\boxed{7}\boxed{9}$ Ⓔ None

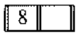


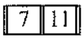
5. After deleting the key 9 from the B+ tree , the **root** of the tree becomes:

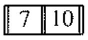
   Ⓐ $\boxed{8}\boxed{}$ Ⓑ $\boxed{7}\boxed{11}$ Ⓒ $\boxed{7}\boxed{10}$ Ⓓ $\boxed{7}\boxed{9}$ Ⓔ None

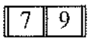


6. After deleting the key 12 from the B+ tree , the **root** of the tree becomes:

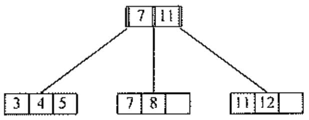
Ⓐ 

Ⓑ 

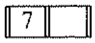
Ⓒ 

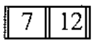
Ⓓ 

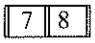
Ⓔ None

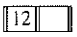


7. After deleting the key 11 from the B+ tree , the **root** of the tree becomes:

Ⓐ 

Ⓑ 

Ⓒ 

Ⓓ 

Ⓔ None

Question 8 . . . . 6 points

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

1. What is the least number of edges a graph with 50 nodes must have to be connected?

   Ⓐ 50. Ⓑ 51. Ⓒ 49. Ⓓ 100. Ⓔ None.

2. When searching for a node (not too far from the source node) in a graph, which of the following is true?

   Ⓐ DFS is more appropriate than BFS. Ⓑ BFS is more appropriate than DFS. Ⓒ DFS is not suitable at all for this case. Ⓓ BFS is not suitable at all for this case. Ⓔ None.

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

A	$\to C \to D$
B	$\to D \to E \to F$
C	$\to A \to F$
D	$\to A \to B$
E	$\to B \to F$
F	$\to B \to C \to E$

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

   (a) (A, D, B, E, F) ___

   (b) (D, B, F, A) ___

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

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

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

   (a) The number of edges in this graph is 6. ___

   (b) The graph can accept new unique edges. ___

   (c) In the graph, (A, C, F, B, A) is a cycle. ___

(d) The graph is connected. ___

3. Which of the following is true for this graph ? Ⓐ The graph is disconnected. Ⓑ The graph has no cycles. Ⓒ The graph is connected and has no cycles. Ⓓ The graph has cycles. Ⓔ None.

4. The BFS traversal of this graph starting from node C is (insert neighbors in the data structure in increasing alphabetic order): Ⓐ C, F, D, B, E, A. Ⓑ C, A, D, B, E, F. Ⓒ C, A, F, D, B, E. Ⓓ C, F, E, B, D, A. Ⓔ None.