QUEENS COLLEGE Department of Computer Science
CSCI 313 First Midterm Exam Spring 2018 04.09.18

Instructor: Alex Ryba

07.45am – 09.00am, Monday, April 09, 2018

Problem 1 *(10 points)*

In (a) and (b) give useful Θ estimates for the function $t(n)$.

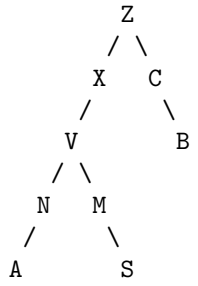
(a) $t(n)$ is the running time for the *preOrder* method applied to a binary tree that contains n nodes.

Answer: $\Theta(n)$

(b) $t(n)$ satisfies $t(n) = 10t(n/2) + (n-1)n(n+1)/6$.

Answer: $\Theta(n^{\log_2 10})$

For the following binary tree:



(c) What is the height of the node that stores `X`?

Answer: 3

(d) What is the depth of the node that stores `X`?

Answer: 1

(e) What is the postorder traversal?

Answer: A N S M V X B C Z

Problem 2 *(10 points)* The following is part of a circular array based model for a `Deque`. The method `addLast` has logic errors and does not operate correctly. Other required methods have been entirely omitted.

```
public class Deque<T> {
  private T data[];
  private int front, rear, size;
  // the next items to be added at front and rear will be placed at indices front and rear

  public Deque() {   // This is correct code.
    data = (T[]) new Object[100];
    front = size = 0; rear = 1;
  }

  public T addLast(T x) throws Exception {  // This has errors.
    ++size;
    if (size > 100) throw Exception("Deque Full");
    data[++rear] = x;
    if (rear > 100) rear = 0;
  }

  // other methods omitted

}
```

(a) Correct the code for `addLast`. Your solution should not use more more than one additional line of code and should be based on that given. It must correct any errors.

Answer:

```
public void addLast(T x) throws Exception {
  if (size >= 100) throw new Exception("Deque Full");
  data[rear++] = x;
  if (rear >= 100) rear = 0;
  ++size;
}
```

(b) Give a title line for each required method that has been omitted. (Credit will be given for the 3 most important methods only.)

Answer:

```
public void addFirst(T x) throws Exception {
public T removeFirst() throws Exception {
public T removeLast() throws Exception {
```

Problem 3 *(10 points)* The *class BNode* implements a generic binary tree node. The class provides the following instance variables and methods.

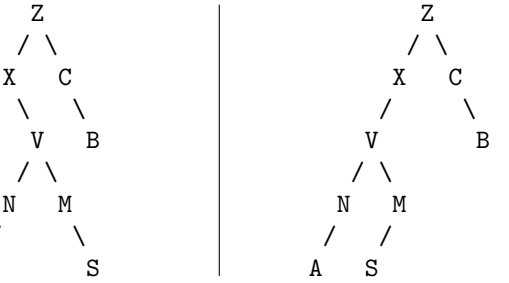
```
public class BNode<T> {
   BNode<T> parent, left, right;
   T data;
   public BNode(T data, BNode<T> parent, BNode<T> left, BNode<T> right) // code omitted
   // getters and setters for all instance variables, code omitted

   public void leftShift()             // to be written

}
```

Write complete code for the method *leftShift* that acts on the subtree of descendants of the `BNode`. For any node `lower` in this subtree that has only a right child, that right child should be moved to become a left child.

For example, if `node.leftShift()` is called at the node storing **X** in the tree of the left hand diagram, the result is the tree in the right hand diagram.



Code longer than 15 lines may be subject to a penalty. Code with a running time worse than $O(n)$ (where n is the number of nodes in the subtree being adjusted) will be subject to a penalty.

Answer:

```
public void leftShift() {
    if (left == null && right == null) return;  // base case
    if (left == null) {
        left = right;
        right = null;
    }
    left.leftShift();
    if (right != null) right.leftShift();
}
```