Prelim 2[Solutions]

1. **Short Answer** [18 pts]

   (a) [4 pts]   A traversal of an expression tree produces the string "$+ * 2\,5 + 1\,3$".

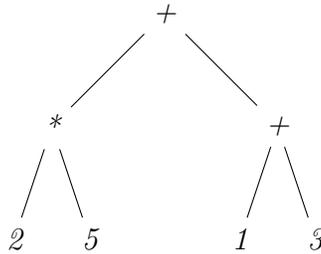   i. What kind of traversal is it?

   > **Answer:**
   >
   > *Preorder; the operator is printed before the operands.*

       ii. What is the result of evaluating the expression?

   > **Answer:**
   > *The string must represent the tree*
   >
   >
   >
   > *This expression represents* $(2 * 5) + (1 + 3)$, *which is* 14.

   (b) [4 pts]   Which of the following must be true (there may be more than one)?

       i. if x.hashCode() == y.hashCode() then x.equals(y)

       ii. if x.hashCode() == y.hashCode() then x == y

       iii. if x.equals(y) then x.hashCode() == y.hashCode()

       iv. if x == y then x.hashCode() == y.hashCode()

   > **Answer:**
   > *The first and second need not be true. The third must be true because otherwise hash table lookups will not find entries that are equal. The fourth must be true because hashCode should return the same result if the object hasn't changed.*

   (c) [3 pts]   A developer wants to use the observer pattern to react when the value of a counter changes. Which of the following steps should the program take *during initialization*? Note: there may be more than one.

       i. Create a new CounterListener object

       ii. Call addCounterListener

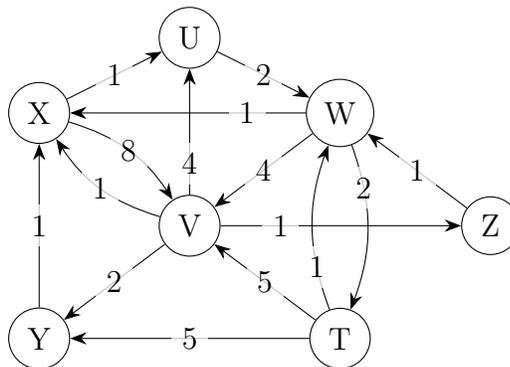   iii. Call counterChanged

   > **Answer:**
   > *The program should create a new listener and add it, but should not call counter changed.*

(d) [7 pts]   Let $f(n) ::= n + n \log n$. Then $f(n)$ is $O(g(n))$ for which of the following g? Note: there may be more than one.

- $g(n) ::= n \log n$ **True**
- $g(n) ::= n$ **False**
- $g(n) ::= 2^n$ **True**
- $g(n) ::= \log n$ **False**
- $g(n) ::= (n + n \log n)/2$ **True**
- $g(n) ::= n + n \log n - 1$ **True**
- $g(n) ::= \sqrt{n + n \log n}$ **False**

2. **Graphs** [18 pts]   Consider the following graph:

(a) [5 pts]   List the nodes in the order they would be visited by DFS, starting from X. When there are multiple valid options, list the lower-letter nodes first. (e.g. if you can visit either V or Y next, visit V next)

Answer:

$X\ U\ W$ QQQQQQ T V Y Z

(b) [5 pts]   List the nodes in the order they would be visited by BFS, starting from X.

Answer:

$X\ U\ V\ W\ Y\ Z\ T$

(c) [5 pts]   List the nodes in the order they would be visited by Dijkstra's algorithm, starting from X. Write down the resulting path lengths.

[TODO — clarification: "settled" instead of "visited"]

Answer:

X	U	W	T	V	Z	Y
0	1	3	5	7	8	9

(d) [3 pts]   Is the graph planar?

Answer:
Yes.

3. **Sorting** [14 pts]   In this question we develop a sorting algorithm using invariants. b is an array of integers; $b[h..k]$ is to be sorted.

- Precondition: b is an array with $0 \le h \le k < b.length$ (nothing is known about the values).
- Postcondition: $b[h..k]$ is sorted.
- Invariant: $h \le i \le k$ and $b[i..k]$ is sorted.

(a) [2 pts]   Give an array diagram that captures this invariant.

Answer:

(b) [2 pts]   Give the initialization statements that make the invariant true.

Answer:

i=k;

(c) [2 pts]   Give the loop condition.

Answer:

i > h; (or i != h;)

(d) [2 pts]   Give a statement that makes progress towards termination (it may violate the invariant).

Answer:

i = i−1;

(e) [3 pts]   Describe in English what must be done in the loop body before part (d) to preserve the invariant. Your answer should be roughly one sentence.

Answer:
Insert $b[i-1]$ into the correct place in $b[i..k]$, moving everything smaller than $b[i]$ one space to the left.

(f) [3 pts]   What is the asymptotic worst-case running time of this algorithm?

Answer:
$O(n^2)$

4. **GUIs** [10 pts]   Consider the program on the next page.

(a) [5 pts]   Draw the GUI resulting from running the program on the following page.

Answer:

(b) [5 pts]   The code doesn't work: nothing happens when the button is pressed. Explain why not, and fix it. Hint: this requires a one-line change. Don't worry if you don't remember the exact names of any swing methods you may need.

Answer:
The label is not updated when the count changes. The code is updated to call label . setText in the actionPerformed method.

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class GUISol extends JFrame {

  /** invariant: label.getText() = count */
  private JLabel label;
  private int    count;

  private class IncrementButton
         extends JButton
      implements ActionListener
  {
    public IncrementButton() {
      super("Increment");
      addActionListener(this);
    }

    /** increment count and update label. */
    public void actionPerformed(ActionEvent e) {
      count = count + 1; label.setText("" + count);
    }
  }

  public GUISol() {
    super("GUISol");

    this.label = new JLabel("0");
    this.count = 0;

    setLayout(new FlowLayout());
    add(new IncrementButton());
    add(label);
    pack();
  }

  public static void main(String[] args) {
    new GUISol().setVisible(true);
  }
}
```

5. **Spanning Trees** [10 pts]   Recall the following definition:

> **Definition:** A spanning tree T of a connected graph G is a subgraph of G containing all the nodes of G and a minimal set of edges that connects them.

Based on this definition, describe a high-level algorithm (as done in lecture) to find a spanning tree of G.

> **Answer:**
>
> *Repeat until no longer possible: Find a cycle in G and delete one edge of the cycle.*
>
> *Alternatively: add edges that don't create cycles until connected.*
>
> *Any correct algorithm is acceptable.*

6. **Heaps** [14 pts]

In this question, you will perform operations on a min-heap in which the values are the priorities. When you change the value of a node, cross out the old value and write the new value next to it. For example, after swapping the root with the left child and then swapping the root with the right child,

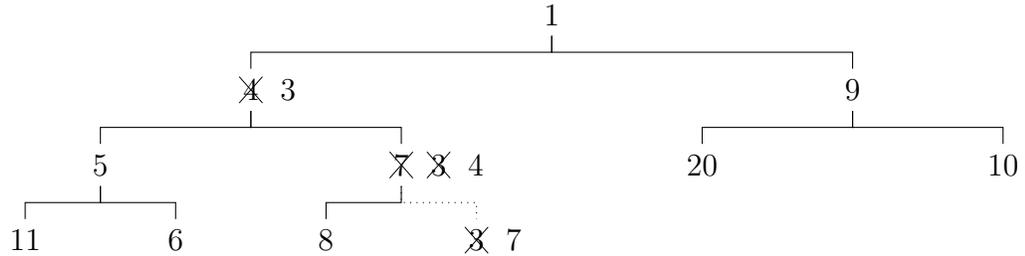**[TODO — there is a typo in the example; should be right then left]**
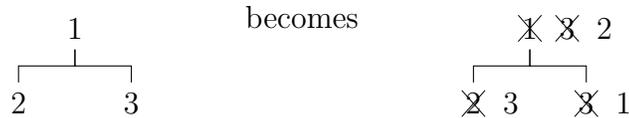
If you make a mistake, be sure that you differentiate between the mistake and the values that you are crossing out!

(a) [7 pts]   Perform poll on the following min-heap:
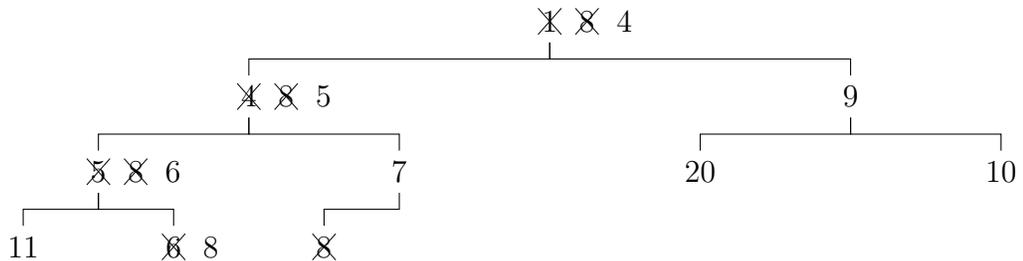
(b) [7 pts]   Perform add(3) on the following min-heap:

1

X 3   9

5   X X 4   20   10

11   6   8   X 7

7. Complexity [16 pts]

(a) [4 pts]   What is the definition of "$f(n)$ is $O(g(n))$"?

Answer:
There exist constants $c > 0$ and $N > 0$ such that for all $n \geq N$, $f(n) \leq cg(n)$.

(b) [6 pts]   Prove that $2n^2 + 2$ is $O(n^3 - 1)$.

Answer:
We would like to show that for all $n > N$, $2n^2 + 2 \leq c(n^3 - 1)$. If $n > 3$ then $n^3 > 3n^2 = 2n^2 + n^2 > 2n^2 + 9$. We can use this to make progress on our desired equation:

$$
\begin{aligned}
2n^2 + 2 &\leq n^3 - 9 + 2 & &\textit{by above, assuming } n > 3 \\
&\leq n^3 - 1 & &\textit{since } -7 < -1 \\
&\leq c(n^3 - 1) & &\textit{if we choose } c = 1
\end{aligned}
$$

Thus, if we choose $N = 3$ and $c = 1$ we see that the desired result holds.

(c) [6 pts]   Give the worst-case and expected case run times of quicksort on an array b of size n. Describe a situation in which it will take the worst-case time.

Answer:
The expected running time is $O(n \log n)$. The worst-case time is $O(n^2)$; this could occur, for example, if b is sorted. In this case, partitioning an array of size k will always split the array into a subarray of size 0 and a subarray of size $k - 1$, so each recursive call to quicksort will be on an input of size $k - 1$; leading to $O(n^2)$ total time.