This is a *closed-book*, *closed-computer*, *open-mind* written exam. You are **not** to discuss the questions on this exam until after it is returned in class.
Write your name, clearly, on your exam.
Answer questions in the spaces provided; if you need additional space, continue on the *back* of an exam page or on *another piece of paper* **after** marking down what question is being answered.
Read the **entire** exam, *carefully*, before you start answering questions. This lets you allot your time appropriately. It also helps you see related questions.
There are a total of 100 points on this exam.

1. (5 points) Give the **definition** of the

2. Consider the Java *regular expression*: `[0-9A-Fa-f]+`. Which of the following strings (quotes are *not* part of the string) would **match** that regular expression? That is, where `Pattern.compile("[0-9A-Fa-f]+").matcher(string).matches()` returns **`true`**.
   (a) (3 points) `""`
   (b) (3 points) `"deadbeef"`
   (c) (3 points) `"Hex numbers are fun!"`
   (d) (3 points) `"xyzzy"`
   (e) (3 points) `"<<< 1aF2dc"`

3. (10 points) Write a `reverseArray` function that takes an array of `String` and an `activeCount` (the number of elements in the array that are actually in use) and reverse the order of the elements.

```
public void reversArray(String [] A, int activeCount) { ... } // your job
```

4. (10 points) Write a `reverseTree` function, part of the `BST` class that swaps left and right children for all nodes, basically reversing the BST property in the tree:

```
public void reverseTree() { reverseTree(root); }
private void reverseTree(BSTNode curr) { ... } // your job
```

5. The following questions refer to a *binary search tree* containing letters, sorted in standard alphabetic order.
   (a) (9 points) **Draw** the BST resulting from inserting the following letters in the given order.

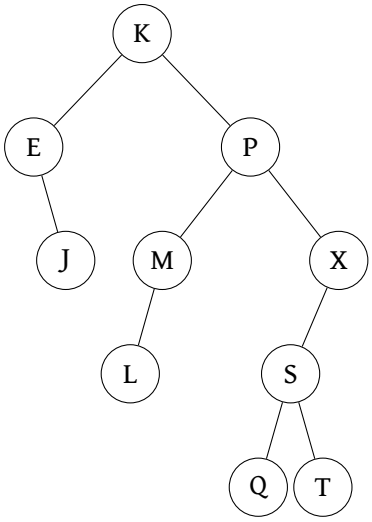
```
M F T Q V B K O D A V S L J
```


Points earned: _____
out of a possible 49 points

(b) (3 points) What is the **height** of the resulting tree?

(c) (3 points) Are there any letters that could be inserted in the tree that would not increase its height? If so, what are they? If not, justify your answer.

6. Consider the following *binary search tree* and a delete method that swaps with a node's *successor* when swapping is necessary.



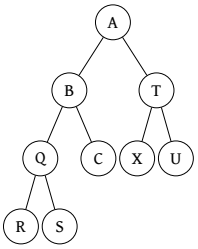
(a) (5 points) How many *leaves* are there in this tree?

(b) (5 points) **Draw** the BST resulting from the deletion of the node labeled `M`.

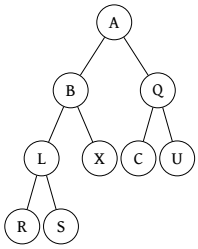
(c) (10 points) **Draw** the BST resulting from the deletion of the node labeled `P` from the *original* (still has `M`) tree.

7. Each of the following trees is either a *binary search tree*, a *min-heap*, a *max-heap*, or none of the above. **Label** each.

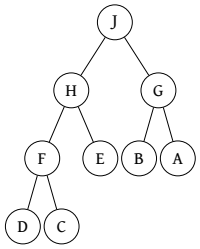
(a) (5 points)



(b) (5 points)



(c) (5 points)



(d) (10 points) Copy the first (by part letter) *heap* into the array below using the technique used to store a binary tree in an array.

Points earned: _____
out of a possible 51 points