

CS 146 Exam 1

Name _____

Grade:

The test will be open book, open notes, 75 minute time limit. Please write your answers on the exam sheet. Ten problems, 10 points per problem. No computers allowed.

1. *Analysis of non-recursive algorithms.*

A repetitive string is one that is formed by concatenating several copies of some smaller string. For example, *ababab* is repetitive because it consists of three repetitions of *ab*.

Consider the problem of determining whether a string is repetitive or not, by means of a Java method *boolean isRepetitive(String x)*.

Examples: The following strings are repetitive: *aaa, abab, abcabcabcabc, giraffegiraffe*. The following strings are not repetitive: *aba, abbbabb, a, ab*.

(a) Write Java code for *isRepetitive*, without using any library functions. Your code should be as efficient as you can make it.

(b) Using Θ -notation, describe the running time needed to solve this problem in terms of the length n of the input string x . Note that the problem asks for Θ -notation, which means you must give a best-possible bound on the running time. Justify your answer.

Answer _____

Justification:

2. (a) Indicate, for each pair of expressions (A,B) in the table below, whether A is O of B. Write “Yes” or “No” in each blank box of the table.

A	B	is A = O(B)?
n^{17}	2^n	
$\lg n$	n^2	
$\lg n + \sqrt{n}$	n	
$\sqrt{n + 5}$	\sqrt{n}	
$\sqrt{n + 5}$	\sqrt{n}	
$n \lg n$	n^2	
\sqrt{n}	$\lg n$	

2 (b) Use Θ -notation to express the worst-case running times of the following algorithms that have been discussed in class or the textbook

- (a) Insertion sort, on an array of length n . _____
- (b) A faster sorting algorithm, on an array of length n . _____
- (c) Binary search, on a sorted array of length n . _____
- (d) Testing an n -digit integer M to see if it is a prime, by dividing it in turn by each number up to \sqrt{M} . _____
- (e) Multiplying two n by n matrices (whose entries are of fixed size, say doubles). _____.

3. Suppose given an initially empty stack, and suppose the following operations are executed. Assume the *add* method pops the top two items from the stack and pushes the sum of those two items.

push(2); push(11); push(3); add(); multiply(); x = peek(); pop();

- (a) What is the final value of x ?
- (b) What are the final contents of the stack?

4. *Queues*. Suppose a queue is implemented in a fixed array capable of holding 80 entries, using the method discussed in the textbook and in class. Suppose the queue is initially empty, and then objects are put into the queue at the rate of 10 per minute while meantime they are processed and removed from the queue at the rate of 5 per minute. After 120 elements have been added to the queue, which of the following is true?

- (a) You can't add 120 elements to an array holding 80 entries.
- (b) There will be 60 elements in the queue, 40 of them at one end of the array and 20 at the other.
- (c) There will be 60 elements in the queue, all in one contiguous segment of the array.

5. *Linked lists*. The class *StringList* has these members:

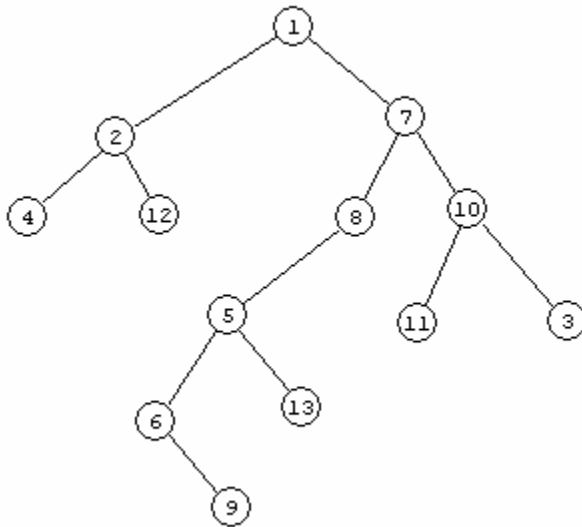
```
{ String key;  
  StringList next;  
}
```

Write a static method *int count(StringList x)* that returns the number of elements in the list.

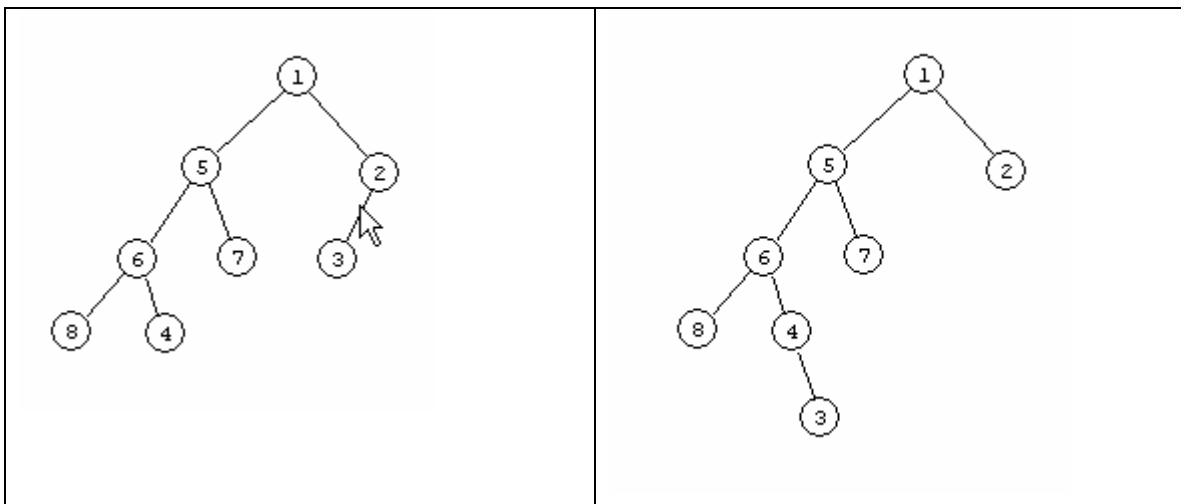
6. Write a method *StringList delete(String needle, StringList haystack)* that deletes the first node containing *needle* from list *haystack*, if any, and returns the (head of) the list after deletion. No new space is allocated, and the input list will be altered. Here *StringList* is the same class used in the previous problem, but the previous problem is not otherwise relevant.

7. *Binary Search Trees.*

The picture represents a binary search tree. **The numbers shown are arbitrary node labels, not numbers representing the contents of the nodes. The contents are not shown.** Please draw the tree that results after the node labeled “7” is deleted, using the textbook algorithm for binary search tree deletion.

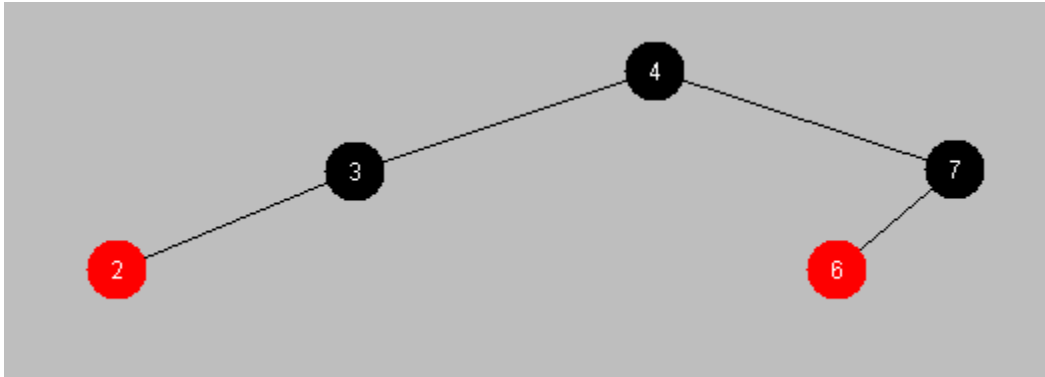


8. *Red-black trees.* For each of the following two trees, either indicate how to color the nodes red and black (use R and B to label the nodes) to make the tree a red-black tree, or explain why that is not possible. In these pictures, as in the applet we used, the NIL leaf nodes are not shown. Ignore the numbers in the diagram, they are meaningless.



9. *Red-black insertion.*

Node 5 is to be inserted into the following red-black tree. In these pictures, as in the applet we used, the NIL leaf nodes are not shown.



Draw the tree after the insertion of node 5, labeling the nodes R and B or coloring them.

10. We studied the Google PageRank algorithm. Consider an example “web” containing three pages A, E, and C in which page A links to E, C links to A, and E and C link to each other. Assume that the pages A, E, and C are indexed as pages 0, 1, and 2 respectively.

(a) What is the B-matrix for this example?

(b) What is the rank order of these pages as produced by the Google algorithm (in descending order of rank)? [You won't get credit for this unless you got part (a) right.]

Circle one of the six possible answers:

- (A,C,E)
- (A,E,C)
- (C,A,E)
- (C,E,A)
- (E,A,C)
- (E,C,A).

Show your calculation if you wish (optional):