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.

The “matching problem” is to determine whether or not the string haystack contains a given substring needle, and if so, find the first number \( k \) such that the substring of haystack starting at index \( k \) and of the same length as needle is equal to needle. Some solution to this problem is used every time you use “find” in some program to search for text in a document.

(a) Write Java code for `public static int match(String haystack, String needle)`. In case needle does not occur as a substring of haystack, return -1. Write your code without using Java library functions such as `substring` or `equals`, so that you work directly with the characters of the strings.

Examples:
match(“racecar”, “car”) returns 4.
match(“car”, “racecar”) returns -1.
match(“automobile”, “mob”) returns 4.

```java
public static int match(String haystack, String needle)
{
    int n = haystack.length();
    int m = needle.length();
    boolean reject;
    for(int i=0; i+m <= n; i++)
    {
        reject = false;
        for(int j=0; j<m && !reject; j++)
        {
            if(haystack.charAt(i+j) != needle.charAt(j))
                reject = true;
        }
        if(!reject)  return i;
    }
    return -1;
}
```

(b) Using \( \Theta \)-notation, describe the running time needed by your code in terms of the lengths \( n \) and \( m \) of the input strings haystack and needle. Note that the problem asks for \( \Theta \)-notation, which means you must give a best-possible bound on the running time of your code.

Answer \( \Theta(nm) \). Two nested loops, one of length \( n \) and one of length \( m \).
(c) Give an example of a worst-case input that demonstrates that the bound you gave cannot be improved.

Answer: haystack = a^{2n} and needle = a^n b. The outer loop runs n times and the inner loop runs n-1 times.

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.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>is A = O(B)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>n\sqrt{3}</td>
<td>2^n</td>
<td>Yes</td>
</tr>
<tr>
<td>\lg n</td>
<td>n^3</td>
<td>Yes</td>
</tr>
<tr>
<td>\lg n + \sqrt{n}</td>
<td>n</td>
<td>Yes</td>
</tr>
<tr>
<td>\sqrt{n} + 5</td>
<td>\sqrt{n}</td>
<td>Yes</td>
</tr>
<tr>
<td>\sqrt{n} + 5n</td>
<td>\sqrt{n}</td>
<td>No</td>
</tr>
<tr>
<td>n \lg n</td>
<td>n^2</td>
<td>Yes</td>
</tr>
<tr>
<td>n^2 + n^3</td>
<td>n^3</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2 (b) Use \( \Theta \)-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 \( n \). Answer: \( \Theta(n^2) \)
(b) Binary search, on a sorted array of length \( n \). Answer: \( \Theta(\lg n) \)
(c) Finding all factors of an \( n \)-digit integer \( M \), by dividing it in turn by each number up to \( \sqrt{M} \). Answer: \( \Theta(2^{n/2}) \)
(d) Adding two \( n \) by \( n \) matrices (whose entries are of fixed size, say doubles). Answer: \( \Theta(n^2) \)

3. Stacks. 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.

\( \text{push}(2); \text{push}(12); \text{push}(6); \text{add}(); \text{multiply}(); \ x = \text{peek}(); \ \text{pop}(); \)

(a) What is the final value of \( x \)? 36
(b) What are the final contents of the stack? empty

4. Queues. Suppose a queue is implemented in a fixed array capable of holding 100 entries, using the method discussed in the textbook and in class, and which you were supposed to use to implement your homework problem \( \text{FixedSizeQueue} \). 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 100 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.

*Answer: (b).* The head and tail of the queue both move along, but the tail moves twice as fast and goes “off the end and around to the beginning”. So after 120 elements have been added, the tail is at index approximately 20 and the head at index approximately 60.

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

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

Write a public static method `int count(String needle, StringList haystack)` that returns the number of elements in the list that are equal to `needle`.

```java
public static int count(String needle, String haystack)
{ StringList marker;
    int ans = 0;
    for(marker = haystack; marker != null; marker = marker.next)
    { if(marker.key.equals(needle))
        ++ans;
    }
    return ans;
}
```

6. *BigMatrix.* This problem refers to your homework `BigMatrix`. Draw a linked-list diagram showing how the following matrix would be represented as a `BigMatrix` object.

```
1  0  0  3  0
0  2  1  0  4
3  0  0  0  2
5  0  6  0  0
```

*Answer:*

```
(0,1) → (3,3)
(1,2) → (2,1) → (4,4)
(0,3) → (4,2)
(0,5) → (2,6)
```
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 root node is deleted, using the textbook algorithm for binary search tree deletion.

Answer: The successor of 12 is 14, so 14 replaces 12 and the node that originally contained 14 is spliced out.

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.
Answer: The first tree can be colored. To get the black-height of 2 right, node 4 has to be red. On the right side we will need black height 2, so node 2 will have to be black. Then nodes 5, 3, and 8 have to be red and nodes 6 and 7 are black. The second tree can’t be colored correctly, since as before we need node 4 to be red, so the black height of the root has to be 2, but that isn’t possible on the long path on the right.


Node 12 is to be inserted into the following red-black tree. In these pictures, as in class and as in the applet linked from the course web page, the NIL leaf nodes are not shown.
Draw the tree after the insertion of node 12, labeling the nodes R and B or coloring them. In case it’s hard to tell when the exam is printed, 10 and 14 are red and the other nodes are black in this picture.

First 12 is the left child of 14. Since it doesn’t have a red uncle, and its parent is the left child of its parent 16, we right-rotate the grandparent 16, producing the answer shown.
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 E links to C and A. Assume that the pages A, C, and E are indexed as pages 0, 1, and 2 respectively.

(a) What is the B-matrix for this example? First we note nLinks = (1, 2, 2). So the entries in column 0 will be (0, 0, 1) (these are the links from A), and in column 1 we have the links from C, but divided by 2, so we get (0.5, 0, 0.5), and in column 2 we have the links from E, divided by 2, so we get (0.5, 0.5, 0). Thus the B matrix is

\[
\begin{bmatrix}
0 & 0.5 & 0.5 \\
0 & 0 & 0.5 \\
1 & 0.5 & 0
\end{bmatrix}
\]

(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.]

(E, A, C)

Show your calculation if you wish (optional):

B times (1, 1, 1) is (1, ½, 3/2) giving rank order EAC
B times (1, ½, 3/2) is (1, ¼, 5/4), giving rank order EAC, the same as before, so that’s the answer.