1. [15 points] Here are two methods. Give a worst-case Big-Oh analysis of their performance in terms of the argument n. In each case give some justification or explanation of your answer.

```java
public static int A(int n) {
    int result = 1;
    for (int i = 0; i < n; i++) {
        int s = 0;
        for (int j = 0; j < 10; j++)
            s += result;
        result = s;
    }
    return result;
}

public static int B(int n) {
    int result = 0;
    for (int i = 2; i <= n; i++) {
        boolean ok = true;
        for (int j = 2; j < i; j++)
            if (i%j == 0)
                ok = false;
        if (ok)
            result += 1;
    }
    return result;
}
```
2. [15 points] Here are pictures of two different linked list implementations (A) is singly-linked, (B) is doubly-linked.

Here are some operations. For each give a worst-case Big Oh analysis of the running time on a list of size n:

<table>
<thead>
<tr>
<th>Operation</th>
<th>ArrayList</th>
<th>Singly-linked list</th>
<th>Doubly-linked List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert at the front</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert at the end</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete the first element</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete the last element</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get the middle element</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. [10 points] Describe (in two or three English sentences) a sorting algorithm that will work for linked lists (you can choose whether these are singly-linked or doubly-linked). “Dump it into an array, sort the array and dump it back” is not an acceptable answer. Give a Big-Oh upper bound for the running time of your algorithm on a list of size n.
4. [10 points] What are the two cardinal rules for recursion?

[Note for 2014 – I would probably ask this differently now.]

5. [10 points] What is the difference between an abstract class and an interface? Give an example where an interface is useful.
6. [20 points] Here is a recursive method

```java
public static int H(int n) {
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else if (n%2 == 1)
        return 1 + H(n+1) + H(n-1);
    else
        return 2*H(n/2);
}
```

Give a dynamic programming version of this that eliminates the inefficiencies. If you add any arguments to the function write a sentence that says how they are initialized.

[Note for 2014 – Last year I usually included the table for a dynamic program as an argument to the method. This year I have generally made the table be a class variable rather than an argument. Either way, if you use a table you should say how it is initialized.]
7. [20 points] Draw a picture of a linked implementation of a Queue of base type Integer.

   a) What class variables will your Node class contain?

   b) What class variables will your Queue class contain?

   c) Give code for the method `dequeue()`. This should remove and return the element currently at the head of the queue.