Computer Science 136

Professor Bruce

Midterm Examination

March 19, 1997

Question	Points	Score
1	20	____
2	15	____
3	10	____
4	21	____
5	14	____
6	7	____
TOTAL	87	____

Your name (Please print)

____

I have neither given nor received aid on this examination.

____

(sign here)

1. A two-dimensional array `mat` with n rows and n columns is said to be symmetric if for all $0 \le i,j \le n$, `mat[i,j] = mat[j,i]`. For example, the following is a symmetric 3 × 3 array of integers: $\begin{bmatrix} 1 & 2 & 6 \\ 2 & 3 & 8 \\ 6 & 8 & 5 \end{bmatrix}$. Because of the symmetry we can just store the elements on and below the diagonal. Thus if we keep $\begin{bmatrix} 1 & \_ & \_ \\ 2 & 3 & \_ \\ 6 & 8 & 5 \end{bmatrix}$ we can easily reconstruct the missing entries. Because each row is now a different length, we save the remaining entries in a one-dimensional array with 6 elements: [1,2,3,6,8,5]. The location of $a_{i,j}$ in a list representing an n by n symmetric array can be quickly computed as follows:

$$\texttt{index(i,j)} = \begin{cases} \dfrac{\texttt{i * (i + 1)}}{2} + \texttt{j,if i >= j} \\ \dfrac{\texttt{j * (j + 1)}}{2} + \texttt{i, otherwise} \end{cases}$$

Thus the element 8 in mat[2, 1] will be given subscript index(2,1) = 2*3/2+1 = 4 in the one-dimensional representation.

Note that the one-dimensional array representing an n by n symmetric array will need to hold only n(n+1)/2 elements, only slightly more than half of the original n^2 elements.

Please write a class `SymmetricArray` for symmetric two-dimensional arrays. The representation should include a field `numRows` which keeps track of the number of rows (and columns) of the two-dimensional array (remember the number of rows and columns will always be the same for a symmetric array), as well as a field `elts`, a one dimensional array of objects in the lower diagonal of the array (as in the example above).

You should include a constructor for the class which takes an integer parameter `numRows` representing the number of rows and columns of the symmetric array and creates a corresponding one-dimensional array filled with `null`s.

You need only include the following methods:
A function
```
getRows()
```
which returns the number of rows (or columns) in the two-dimensional array, a function
```
getEltAt(int row,int col)
```
which returns the element of the two-dimensional array in the `(row,col)` position, and a procedure
```
setEltAt(Object value, int row, int col)
```
which inserts `value` at a location corresponding to `(row,col)`.

You need not include pre- and post-conditions.

Please write your answer on the following page. You may use the back of the page as well if need be.

Answer to #1:

2. Suppose we wish to extend the capability of the class `LinkedList` to add a new function which prints out all elements of the list (i.e., the value fields of all elements) using `System.out.println`. You may assume that if `elt` is of type `Object` then `System.out.println(elt)` prints out a description of `elt` (this isn't quite accurate, but will simplify the problem.

   Define a subclass (extension) of `LinkedList`, `PrintableLinkedList` which adds this new method, `printList()`, while including all of the old methods of LinkedList. Recall that `LinkedList` has fields:

   ```
   LinkedListElement head;
   int count;
   ```

   while `LinkedListElement` has fields:

   ```
   Object value;
   LinkedListElement head;
   ```

   Your solution should be a complete and legal class definition in Java.

3. The following is a recursive function to find the largest element in an array with subscripts from 0 to last:

```
/*
   pre:  0 ≤ last < floatArray.length
   post : Returns largest real in floatArray[0..last]
*/
public double maxArray(float[] floatArray, int last)
{
   double biggestSoFar ;
   if (last == 0)
      return floatArray[0];
   else
   {
      biggestSoFar = maxArray(floatArray,last-1);
      if (biggestSoFar >= floatArray[last])
         return biggestSoFar;
      else
         return floatArray[last];
   }
}
```

a. What two things must one prove in order to prove that this function is correct (i.e. meets its postcondition)? You need not give the proof itself, just state the two statements which must be proven. Be sure to state any hypotheses which are allowed to be assumed for the proof. Note: The two things you list must be statements about this particular method, not general statements about induction!

b. What is the complexity of the function if `last` = n-1 (i.e., there are n elements in the array being searched)?

4. Short answers:

a. In your Josephus assignment(s), you provided two implementations of the interface Circular. One was the class CircularVector (using a Vector as a field) and the other was CircularJosList (using a circular doubly-linked list as a field). Please explain briefly why I bothered to define the interface. That is, what would have been more difficult or troublesome if we did not have it. (Hint: Though your memory may be a bit rustier on this, we did something similar in your bouncing ball program where the bouncing object had a declared type of Renderable, where Renderable was an interface.)

b. Explain in one or two sentences why the Vector implementation of <u>stacks</u> avoids the disadvantages of the Vector implementations of <u>lists</u> (in terms of the <u>time</u> complexity of operations). Don't just provide the different time complexities, explain why they exist.

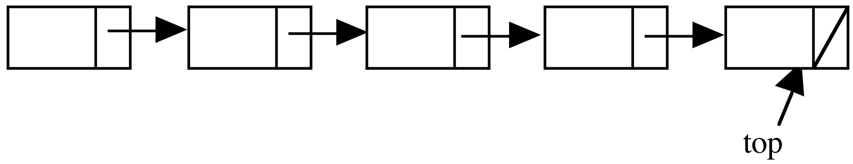
c. Explain briefly why there is a difference in the worst-time and average-time complexity of adding an element to the end of a Vector. Please note what those times are in big-"O" notation.

d. What are the advantages and disadvantages of using a Vector over using a one-dimensional array?

e. Circle the best completion for the following sentence: The main advantage of a circular representation of a linked list over that with pointers to each end of the list is:

   i. It saves space by reducing the pointers into the list from two to one.
   ii. It saves time by eliminating some checks for special cases.

f. The following picture represents a proposed implementation of stacks. Please explain briefly why this would be a bad choice for stacks.



g. We noted in class that the Maze program using stacks could be replaced by a recursive version which did not use an explicit stack data structure. Explain briefly why we could get away without the stack.

5. Provide the complexity of the following operations (in big-"O" notation) if the data structure has n items. Provide the worst case for times in all cases

a. Binary search of an ordered array: ___________

b. Linear search of an ordered array: ___________

c. In the array representation of stacks, the operation push: ___________

d. In the linked list implementation of stacks, the operation push: ___________

e. In the circular singly-linked list implementation of lists, the operation `addToTail`: ___________

f. In the circular singly-linked list implementation of lists, the operation `removeFromTail`: ___________

g. In the Vector implementation of lists, the operation addToHead: ___________

6. Applets and event-driven programming

a. Java programmers are instructed never to directly call the method paint(Graphics g) in their programs, yet the user typically places important code for displaying images in this method. When and how is this method called?

b. Suppose you wish to execute the parameterless method `runProg` when the user clicks on a button represented by variable `startButton` on your subclass of Applet. Please write the method of your subclass which would handle the event and call `runProg`. Be sure to use the proper name and parameters so that it will be called after the click.