CS100J Nov 04, 2003

Arrays. Reading: 8.5

We look at standard algorithms on arrays

The website has about 30 exercises dealing with loops, with answers.

Have your worked on assignment A6 yet? Don't procrastinate!

Quiz on Thursday, on three algorithms that we present today.

More than anything else, mathematics is Method.
Morris Kline.

Quiz on Thursday

I will ask you to write either

- **linear search**,
- **finding the minimum**, or
- **binary search**.

Based simply on that, you write the precondition (if any), postcondition, and invariant, and then develop the algorithm, using the four loopy questions.

You don't memorize the code. Instead, you memorize what the algorithm does (given by the pre- and post-conditions) and develop it invariant and the Java code from there.

To practice for the quiz, practice writing it all down! Don't just read, do. Get a blank piece of paper and go through the whole process. Write the pre- and post-conditions, develop the invariant, and develop the code.

AT THE END OF THESE SLIDE, WE SUMMARIZE THE THREE ALGORITHMS.

- The rest of the lecture will use the blackboard, because that is the easiest way to present the ideas. Pictures of arrays are easier to draw there. Moreover, programming is a dynamic activity, and the sense of development is easier to achieve on the blackboard.

- If you have trouble with this material, listen also to the lectures on the CD!

• Linear search. Vague spec.: find first occurrence of v in b[h..k].

Better spec.: Store an integer in i to truthify:

postcondition: (0) v is not in b[h..i-1]

(1) Either i= k or v = b[k]

invariant: v is not in b[h..i-1]

• Finding the min. Vague spec.: Find the minimum of b[h..k]

Better spec.: Precondition: h <= k (because an empty set of values has no minimum)

Store in I to truthify:

postcondition: b[m] is the minimum of b[h..k] (and it is the first occurrence of the minimum)

invariant: b[m] is the minimum of b[h..t-1] (and it is the first occurrence of the minimum)

• Binary search: Vague spec: Look for v in sorted array segment b[h..k].

Better spec:

Precondition: b[h..k] is sorted (in ascending order).

Store in i to truthify:

postcondition: b[h..i] <= v and v < b[i+1..k]

invariant: b[h..i] <= v and v < b[j..k]