CS1110 6 November 2008

Algorithms on arrays **Reading: 8.3–8.5**

The searching, sorting, and other algorithms will be on the course website, along with a JUnit testing class for them.

Please punctuate this:

Dear John, I want a man who knows what love is all about you are generous kind thoughtful people who are not like you admit to being useless and inferior you have ruined me for other men I yearn for you I have no feelings whatsoever when we're apart I can be forever happy will you let me be yours

Gloria

1

A neat example of the ambiguity of English! We try to use English properly and precisely, but ambiguity tends to creep in because of difference in cultures in which people grow up and simply because of differences of opinion.

I want a man who knows what love is all about. You are generous, kind, thoughtful. People who are not like you admit to being useless and inferior. You have ruined me for other men. I yearn for you. I have no feelings whatsoever when we're apart. I can be forever happy -- will you let me be yours? Gloria

I want a man who knows what love is. All about you are generous, kind, thoughtful people, who are not like you. Admit to being useless and inferior. You have ruined me. For other men, I yearn. For you, I have no feelings whatsoever. When we're apart, I can be forever happy. Will you let me be?
Yours, Gloria .

2

Horizontal notation for arrays, strings, Vectors

	0	k	b.length
b	<= sorted	>=	

Example of an assertion about an array b. It asserts that:

1. b[0..k–1] is sorted (i.e. its values are in ascending order)
2. Everything in b[0..k–1] is ≤ everything in b[k..b.length–1]

	0	h	k
b			

Given the index h of the First element of a segment and the index k of the element that Follows the segment, the number of values in the segment is k – h.

b[h .. k – 1] has k – h elements in it.

h h+1

(h+1) – h = 1

3

Partition algorithm: Given an array b[h..k] with some value x in b[h]:

P: b — h: x | ? — k

Swap elements of b[h..k] and store in j to truthify P:

Q: b — h: <= x | j: x | >= x — k

change: b [h] 3 5 4 1 6 2 3 8 1 [k]

into b [h] 1 2 1 3 [j] 5 4 6 3 8 [k]

or b [h] 1 2 3 1 3 [j] 4 5 6 8 [k]

x is called the pivot value.

x is not a program variable; x just denotes the value initially in b[h].

4

Linear search (for value known to be in array)

Vague spec.: Find first occurrence of v in b[h..k-1], which is known to be there.
Better spec.: Store an integer in i to truthify postcondition Q:

Q: 1. v is not in b[h..i-1]
2. v = b[i]

precondition P: b — h: v is in here — k

postcondition Q: b — h: v not here | i: v | ? — k

Can't simply combine P and Q because position of v not known initially. But we can just delete value v from Q:

invariant Q: b — h: v not here | i: v is in here — k

5

Linear search

Vague spec.: Find first occurrence of v in b[h..k-1].
Better spec.: Store an integer in i to truthify postcondition Q:

Q: 1. v is not in b[h..i-1]
2. i = k OR v = b[k]

P: b — h: v is in here — k

Q: b — h: v not here | i: v | ? — k

OR

b — h: v not here — i, k

6

Binary search: Vague spec: Look for v in **sorted** array segment b[h..k].
Better spec:
Precondition P: b[h..k] is sorted (in ascending order).
Store in i to truthify:
Postcondition Q: b[h..i] <= v and v < b[i+1..k]

Below, the array is in non-descending order:

P: b [h … k] : ?

Q: b [h … i] : <= v, [i+1 … k] : > v

Called binary search because each iteration of the loop cuts the array segment still to be processed in half

7

How many iterations does binary search make?

Suppose k-h is a power of 2.

How many iterations does binary search perform? Figure it out, as a function of k-h, and be ready to tell us on Tuesday.

p	s^p
0	2^0 = 1
1	2^1 = 2
2	2^2 = 4
3	2^3 = 8
4	2^4 = 16
5	2^5 = 32
6	2^6 = 64
…	…

8

Reversal: Reverse the elements of array segment b[h..k].

precondition P: [h … k] : not reversed

postcondition Q: [h … k] : reversed

Change: b [h … k] : 1 2 3 4 5 6 7 8 9 9 9 9

into b [h … k] : 9 9 9 9 8 7 6 5 4 3 2 1

9

Check whether two arrays are equal

```
/** = “b and c are equal” (both null or both contain
        arrays whose elements are the same) */
public static boolean equals(int[] b, int[] c) {

}
```

10