CS100J Nov 04, 2003

Arrays --sorting. Reading: 8.5

Quiz on Tuesday: linear search, binary search, finding the minimum.

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 □

This is a neat example of the ambiguity that English can cause, if not used properly! We try to use English properly and precisely, but ambiguity tends to creep in because of diffierence in cultures in which people grow up and simply because of differences of opinion. Read on!

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 □

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 .

• 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]

Algorithms to be covered today:

- **Partitioning an array segment.** Given an array:

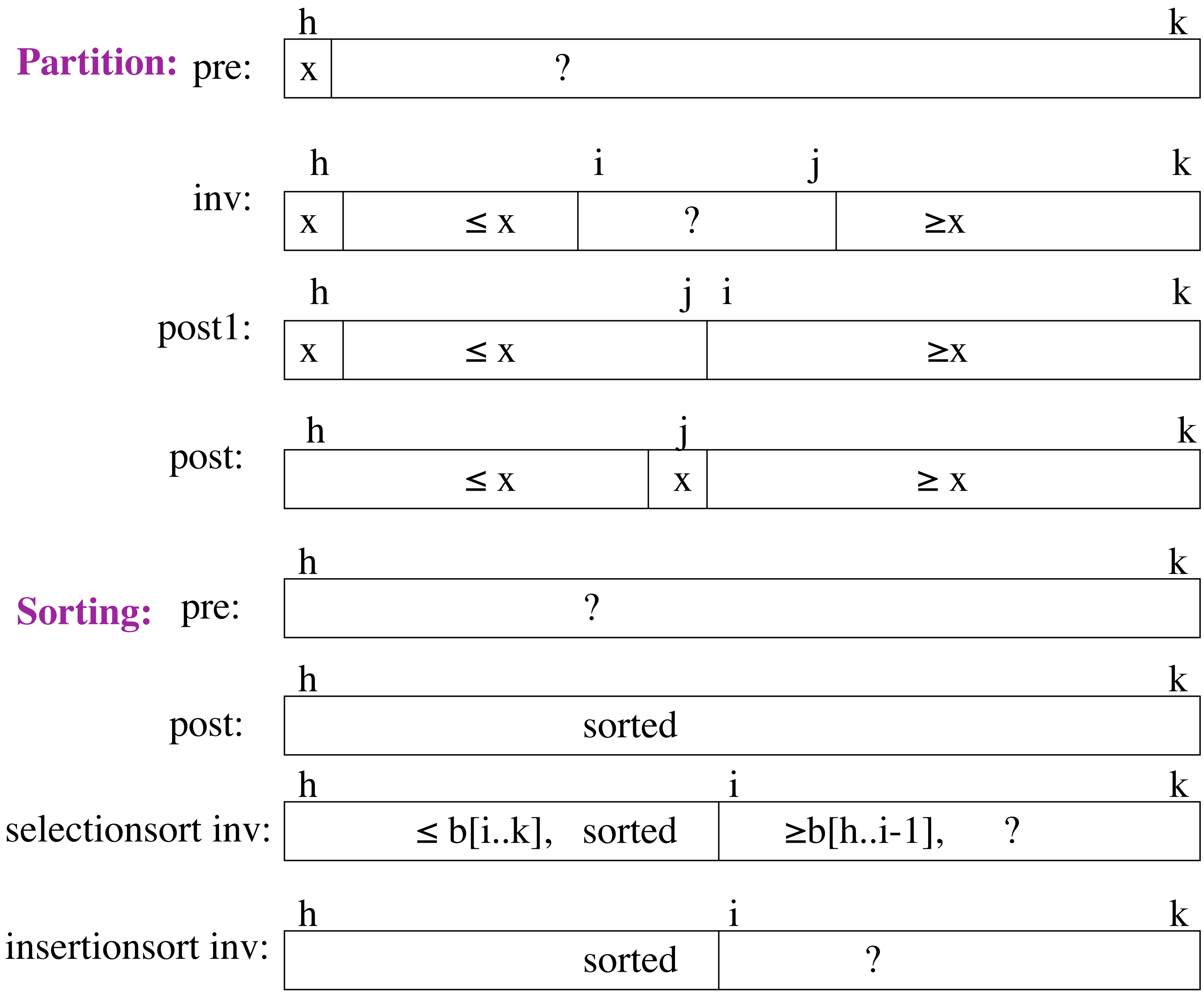
h		k
x	?	

swap its values around so that it looks like this:

h		k
$\le$ x	x	$\ge$ x

- **Selection sort.**

- **Insertion sort.**


Partition: pre:
h k
x ?
inv:
h i j k
x ≤ x ? ≥x
post1:
h j i k
x ≤ x ≥x
post:
h j k
≤ x x ≥ x
Sorting: pre:
h k
?
post:
h k
sorted
selectionsort inv:
h i k
≤ b[i..k], sorted ≥b[h..i-1], ?
insertionsort inv:
h i k
sorted ?