CS100J   01 April 2008
More on arrays: Sorting: insertion- selection- quick- sort

Do exercises on pp. 311-312 to get familiar with concepts and develop skill. Practice in DrJava! Test your methods!
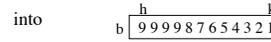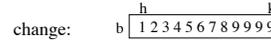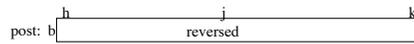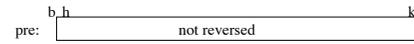
Haikus (5-7-5) seen on Japanese computer monitors

Yesterday it worked.
Today it is not working.
Windows is like that.

A crash reduces
Your expensive computer
To a simple stone.

Three things are certain:
Death, taxes and lost data.
Guess which has occurred?

Serious error.
All shortcuts have disappeared.
Screen. Mind. Both are blank.

The Web site you seek
Cannot be located, but
Countless more exist.

Chaos reigns within.
Reflect, repent, and reboot.
Order shall return.

1

Reversing array segment b[h..k]

2

Remove adjacent duplicates

change:

0											n
b	1	2	2	4	2	2	7	8	9	9	9 9

into

0					h		n	
b	1	2	4	2	7	8	9	8 9 9 9 9

don't care what is in b[k+1..n]
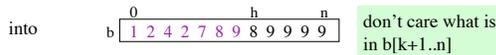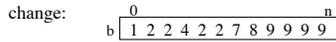
postcondition:

b[0..h] = initial values in b[0..n] but with adj dups removed

pre: b

0	n
?	

post: b

0	h	n
orig values of b[0..n], no adj dups	? (unchanged)	

3
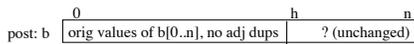
Sorting: "sorted" means in ascending order

pre: b

0	n
?	

post: b

0	n
sorted	

insertion sort
inv: b

0	i	n
sorted	?	

```
for (int i= 0;  i < n;  i= i+1) {
    Push b[i] down into its sorted
        position in b[0..i];
}
```

					i	
2	4	4	6	6	7	5

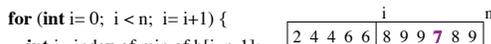|					i
2	4	4	5	6	6	7
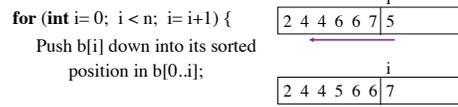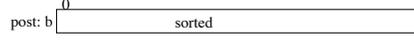
Iteration i makes up to i swaps. In worst case, number of swaps needed is 1 + 2 + 3 + … (n-1) = (n-1)*n / 2.

Called an "n-squared", or n^2, algorithm.

4

---

pre: b

0	n
?	

post: b

0	n
sorted	

insertion sort
invariant: b

0	i	n
sorted	?	

Add property to invariant: first segment contains smaller values.

selection sort
invariant: b

0	i	n
≤ b[i..], sorted	≥ b[0..i-1], ?	

```
for (int i= 0;  i < n;  i= i+1) {
    int j= index of min of b[i..n-1];
    Swap b[j] and b[i];
}
```

					i				n
2	4	4	6	6	8	9	9	**7**	8 9

					i				n
2	4	4	6	6	**7**	9	9	**8**	8 9

Also an "n-squared", or n^2, algorithm.

5

```
/** Sort b[h..k] */                          Quicksort
public static void qsort(int[] b, int h, int k) {
    if (b[h..k] has fewer than 2 elements)
        return;
    int j= partition(b, h, k);
    // b[h..j-1] <= x < b[j+1..k]
    // Sort b[h..j-1]  and  b[j+1..k]
    qsort(b, h, j-1);
    qsort(b, j+1, k);

}
```
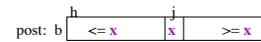
To sort array of size n. e.g. 2^{15}
Worst case: n^2 e.g. 2^{30}
Average case:
        n log n.   e.g. 15 * 2^{15}
                              2^{15} = 32768

pre: b

h	k
x	?

j= partition(b, h, k);

post: b

h	j	k
<= x	x	>= x

6

1

Sir Tony Hoare,   Fritz Bauer
Quicksort author   My advisor

Pioneer in computing in the 50's, 60's. Developer of the historical computing section of the Deutsches Museum

Marienplatz, Munich

Keypunch machine. In the 1960s and early 1970s, a program was punched on such cards. A card deck containing a program to be run would be taken out to the computer, near the Ithaca airport, and run. Get output back 3-4 hours after handing in the card deck.

Zuse built the first working computer in the late '30s, the Z3, completed in 1941. The original machine no longer exists.

Konrad Zuse's Z4

8

2