Scarborough College
University of Toronto
Fall 1997

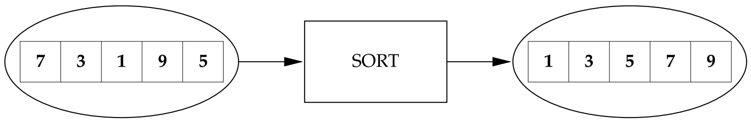
CSC A06F: Introduction to Computer Programming
Lecture notes 6:
Sorting

Reading:

- Lewis and Loftus, chapter 6 and section 13.1.

What is sorting?

Sorting is to rearrange a list of items into their proper order.



There are many methods of sorting.

- Some are more efficient (i.e., faster) than others.
- Some are easy to understand; some aren't so obvious.

Selection sort

Basic Idea: put each item in its correct position, one at a time.

How to do it (for numbers):

- Find the smallest number in the list. Swap it with the first number in the list. (Now the smallest number is in the correct position)
- Find the smallest number in the rest of the list (not including the first number). Swap it with the second number.
- Move to the next smallest number, and repeat.

Example

Consider this list:

At beginning: 7 3 1 9 5

After 1 pass:

After 2 passes:

After 3 passes:

After 4 passes:

After 5 passes:

Sorting an array

Assume we have set up an array `nums` of n integers:

```
int[] nums = { 7, 2, 4, 3, 9, 6, 8, 1, 5 };

// call the sorting method
Sort.selection_sort (nums);
```

How many passes does `selection_sort` have to make?

What does each pass do?

What structure will the `selection_sort` method have?

A selection sort method

```
class Sort {

   public static void selection_sort (int[] list) {

      for (int pass=0; pass < list.length; pass++) {

         // look for index of smallest item in list
         int min = pass;
         for (int i=pass; i< list.length; i++) {
            if (list[i] < list[min]) {
               min = i;
            }
         }

         // swap elements
         int temp = list[pass];
         list[pass] = list[min];
         list[min] = temp;
      } // for
   } // selection_sort
} // Sort
```