Selection and Insertion Sort

Deborah A. Trytten

CS 1323/1324

Algorithms

- There are lots of systematic ways of solving most problems
- Figuring out which way is better, or even best, is important in computer science
- Chance for non-CS majors to get a real feel for how CS people think about computing
  - Computing is not just programming!

Sorting Algorithms

- Sorted data allows the use of binary search instead of linear search
- Sorting has a cost (space and time)
  - The cost depends on the details of the algorithm
- Every good software engineer knows at least ten sorting algorithms by heart
  - Most use techniques beyond the scope of this class
- The algorithms we'll examine use nested loops
- Name of each algorithm is important

Selection sort

- Find the smallest element and put it in the first position
- Find the second smallest element and put it in the second position
- Continue this pattern until the entire array is sorted

Example: Selection Sort

- Trace the code by showing swaps for selection sort on this data using a table
  - `int[] data = {9, 4, 2, 1, 7, 8, 3, 6, 5};`
- Array will be partitioned into sorted and unsorted areas
- One value at a time will move from unsorted to sorted

Think Pair Share

- Trace selection sort by showing swaps for this data using a table
  - `int[] data = {1, 4, 7, 9, 5, 3, 2};`

iClicker Question

- Suppose the following are the initial contents of an array: `{2, 4, 6, 8, 5}`
- Which sequence of steps below represents selection sort?

a)

2	4	6	8	5
		5		6
			6	8

b)

2	4	6	8	5
		5	6	
			6	8

c)

2	4	6	8	5
			5	8
		5	6	

d)

2	4	6	8	5
		5		
			6	
				8

Write Code: Selection Sort

- Write the code for selection sort:
  - `void selectionSort(int[] array)`

- Details:
  - What does the outer loop do?
  - What does the inner loop do?
  - Triangular or rectangular?

Insertion Sort

- Array also partitioned into sorted and unsorted areas
- We take the next unsorted data value and place it into the proper position in the sorted data
- Technique is critical—***does not use swaps***
  - Next value to be sorted is placed in ***temporary storage***
  - Larger values shifted to right until proper position found
  - Value copied from temporary storage to proper position in the array
  - Why no swaps? (Note: different than textbook algorithm)

Example: Insertion Sort

- Trace insertion sort in a table with the same data as before:
  - `int[] data = {9, 4, 2, 1, 7, 8, 3, 6, 5};`

- Move 1 value at a time
  1. Copy first value in unsorted part to temp
  2. Shift larger values in sorted part to make room
  3. Copy value from temp back to array

Think Pair Share

- Trace insertion sort with the following data:
  - `int[] data = {1, 4, 7, 9, 5, 3, 2};`

iClicker Question

- Suppose the following are the initial contents of an array: `{2, 4, 6, 8, 5}`
- Which sequence of steps below represents insertion sort?

a)

2	4	6	8	5
		5		6
			6	8

b)

2	4	6	8	5
				8
			6	
		5		

c)

2	4	6	8	5
			5	8
		5	6	

d)

2	4	6	8	5
		5		
			6	
				8

Write Code: Insertion Sort

- Write the code for insertion sort:
  - `void insertionSort(int[] array)`

- Details:
  - What does the outer loop do?
  - What does the inner loop do?
  - Triangular or rectangular?

Comparison: Worst Case

- Computer scientists analyze algorithms by looking at the worst thing that can happen
- What is the worst order for insertion sort?
- What is the worst order for selection sort?
- Consider assignments versus comparisons
- Which is better if the data are partially sorted?
  - Why is this important?