Bubble Sort

Jack Peachey
CS 32

Basic Idea

- Evaluates list or array, switching terms if based on size and index

- Smaller sets "bubble" to the top

- Xerox analogy

- Brute force method, but useful in checking if list is sorted

Picturing the Bubble Sort

BUBBLE SORT

initial arrangement	8	6	1	4	9	2	5	3	0	7
after pass with r = 9	6	1	4	8	2	5	3	0	7	9
after pass with r = 8	1	4	6	2	5	3	0	7	8	9
after pass with r = 7	1	4	2	5	3	0	6	7	8	9
after pass with r = 6	1	2	4	3	0	5	6	7	8	9
after pass with r = 5	1	2	3	0	4	5	6	7	8	9
after pass with r = 4	1	2	0	3	4	5	6	7	8	9
after pass with r = 3	1	0	2	3	4	5	6	7	8	9
after pass with r = 2	0	1	2	3	4	5	6	7	8	9
after pass with r = 1	0	1	2	3	4	5	6	7	8	9
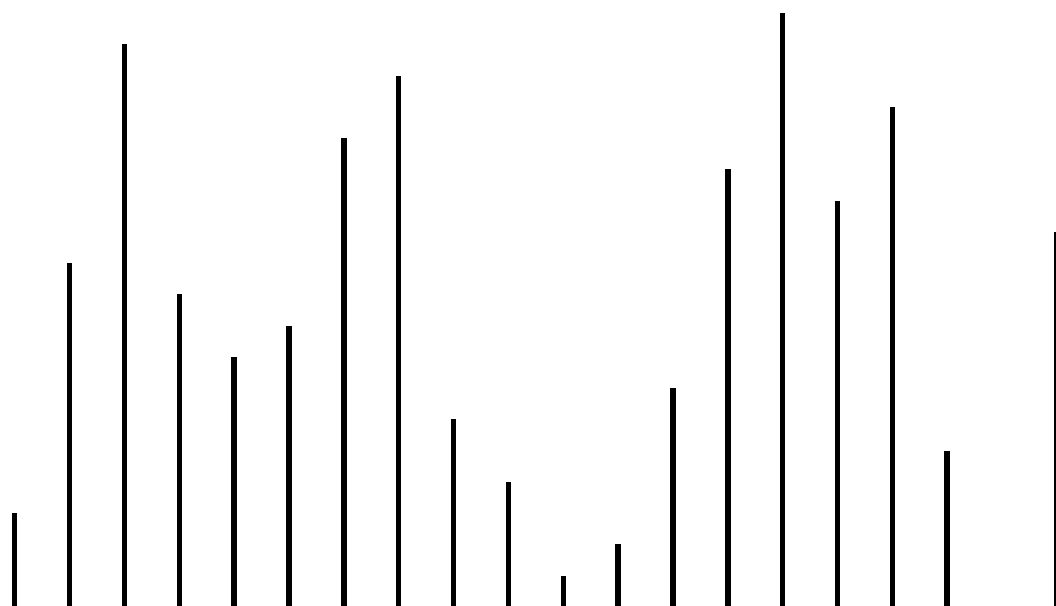
Run Time

- $O(n^2)$ average and worst case

- $O(n)$ best case

Effectiveness

- Runs faster if smaller sets are near the front
- Not dependent on placement of larger sets

What Data Structures does the Bubble Sort need?

Data Structures and Memory Concerns

- Not dependent on any data structures, just a compare and swap method

- Doesn't require dynamic memory, amount used set from beginning

- Runs in a self contained manner

Summary

Sources

- [1]-http://www.algolist.net/Algorithms/Sorting/Bubble_sort

- [2]-http://www.shannarasite.org/kb/kbse26.html

- [3]-http://www.cise.ufl.edu/~mssz/DatStrucAlg/DSAintro.html

- [4]-http://users.informatik.uni-halle.de/~jopsi/dinf204/chap8.shtml