CS 150: Sorting Algorithms

Cynthia Taylor
Oberlin College
April 16th 2014

Sorting

- Before

```
A = [45, 23, 5, 98, 12]
```

- After

```
A = [5, 12, 23, 45, 98]
```

Start with 3 Searching Algorithms

- Selection Sort
- Insertion Sort
- Bubble Sort
- All $O(n^2)$

today

+ 1 more

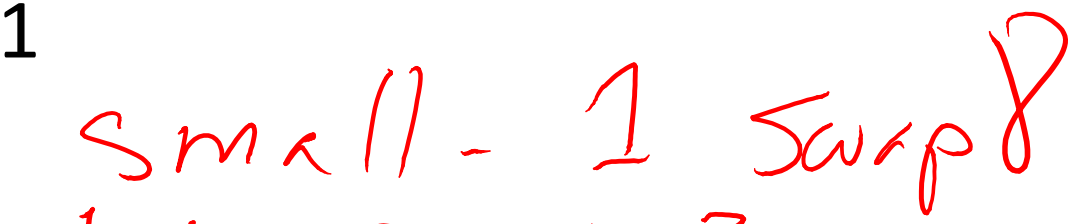
Selection Sort

- List consists of two parts: a sorted part, and an unsorted part
- Initially, our sorted part is empty, and our unsorted part is the whole list

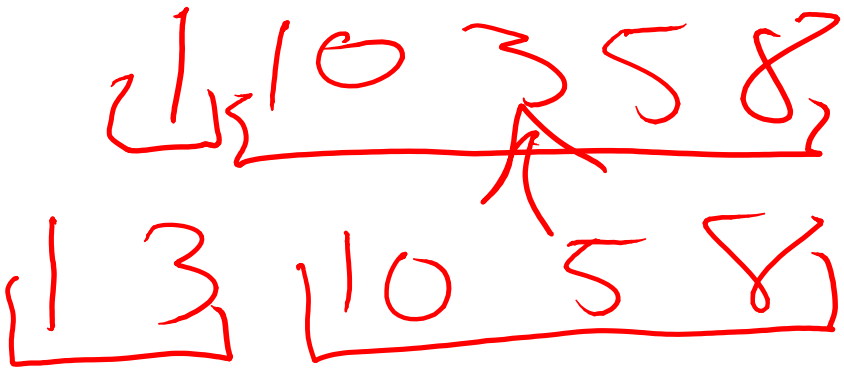
- Then, while our sorted part is not the whole list
    1. Find the smallest remaining value anywhere in the unsorted part, and
    2. Swap this value with the leftmost value in the unsorted part

Selection Sort

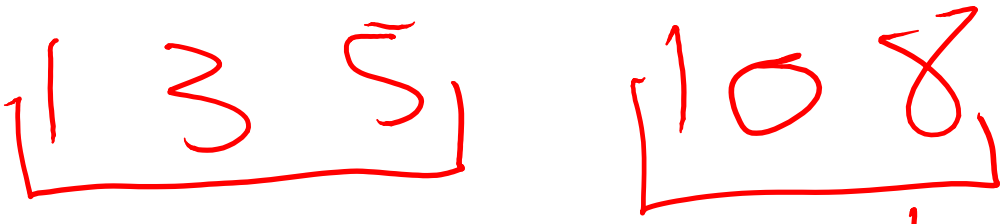
- List: 8 10 3 5 1
- After 1 pass:



- 2 passes:



- 3 passes:



- 4 passes:

Which of the following is true for selection sort?

A. Once a value is placed in the sorted part, it will never move again

B. All values in the sorted part are always less than or equal to all values in the unsorted part

C. Both of the above are true

D. None of the above is true

E. I don’t know

Write Code for Selection Sort

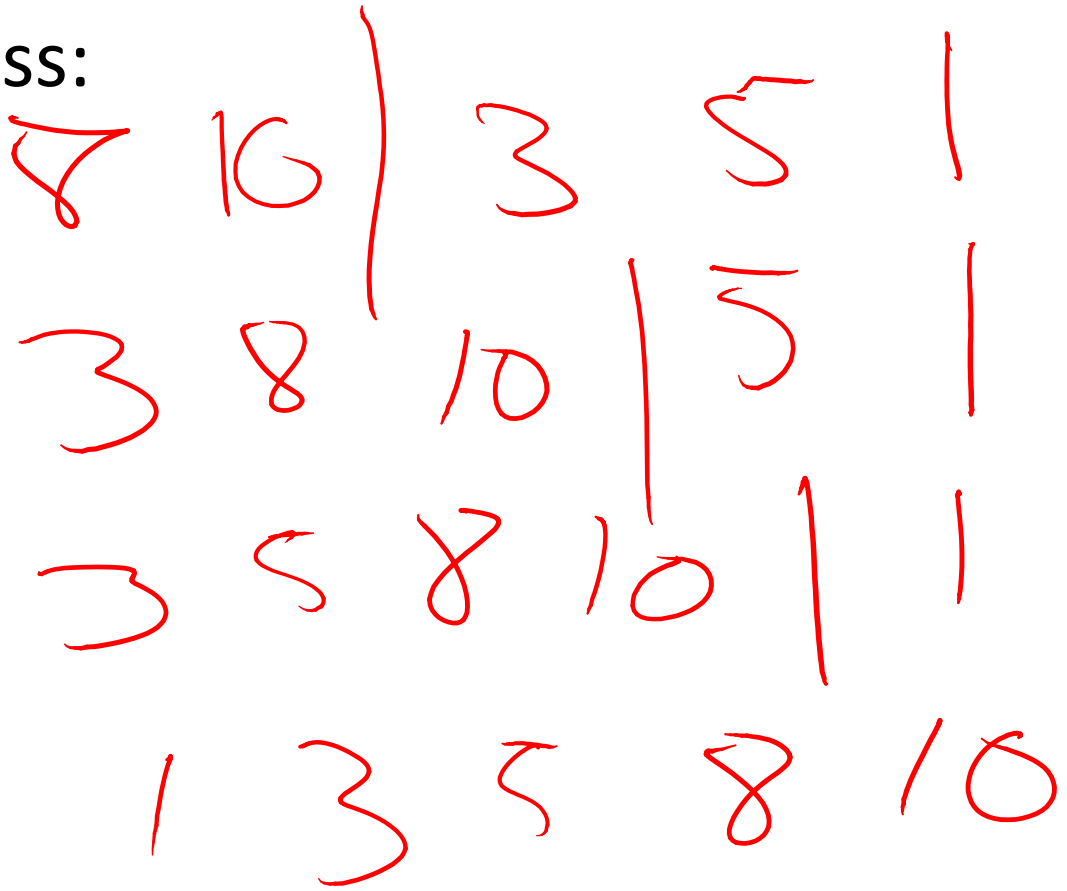
Insertion Sort

- Insertion sort also divides the list into a sorted part (initially empty) and an unsorted part (initially the whole list)

- Then, while our sorted part is not the whole list
    1. Obtain the leftmost value in the unsorted part
    2. Insert this value in its correct place in the sorted part

Insertion Sort

- List: 8 10 3 5 1
- After 1 pass:
- 2 passes:
- 3 passes:
- 4 passes:



Which is true of Insertion Sort?

A. Once a value is placed in the sorted part, it will never move again

B. All values in the sorted part are always less than or equal to all values in the unsorted part

C. Both of the above are true

D. None of the above is true

E. I don't know

[10, 20, 30, 40, 16, 94, 8, 22]

The list above reflects the state of the list after 3 passes of insertion sort.

What will be the list after the next (fourth) pass?

A. [8, 20, 30, 40, 16, 94, 10, 22]

B. [10, 16, 20, 30, 40, 94, 8, 22]

C. [10, 16, 30, 40, 20, 94, 8, 22]

D. [8, 10, 20, 30, 40, 16, 94, 22]

E. I don’t know

Insertion Sort: Complication

- When writing code for insertion sort, we run into a problem.

```
[10, 20, 30, 40, 16]
```

- We know that the 16 should go at index 1
- But we can't just put 16 there, because it would overwrite the 20
- What we do is shift each sorted element to the right until the place for 16 is found

Write Code for Insertion Sort

write shift code

Next Time

- Bubblesort and MergeSort
  - Section 13.3, 5.1-5.2

- Lab 9 – Tuesday at 10 pm