Review:
Bubble Sort and Selection Sort

2 What is sorting?

#include <algorithm>

std::sort(v.begin(), v.end());

- Input: sequence of n comparable values
- Reorder the input to be non-descending.
- Items we wish to sort are called "keys"

- Not here: retain associated information

3 Why discuss sorting?

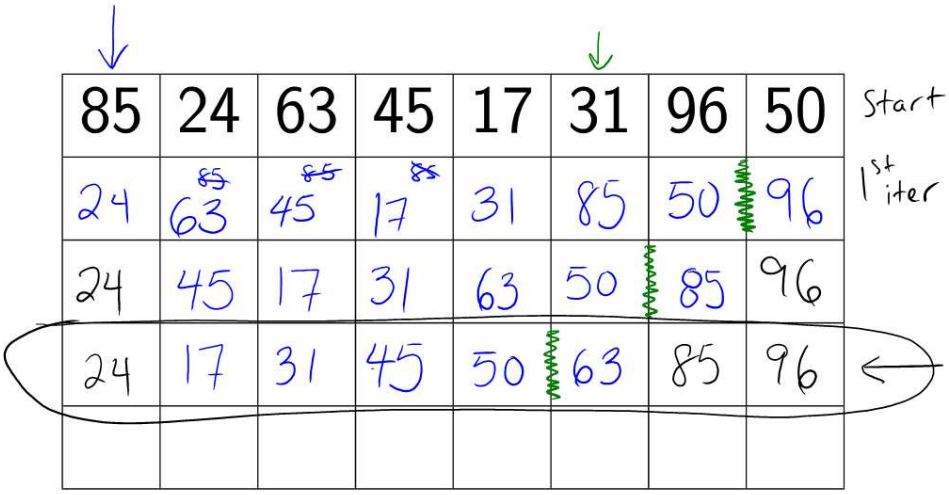
- Standard library has sorting
- Why not use that and move on?

In this class, sorting is:

- a good intro for techniques
- a good intro to comparative algorithms

4 BubbleSort

Idea: Think globally act locally



5 # BubbleSort

```
for i ← 1 to n − 1 do
  for j ← 1 to n − i do
    if A[j + 1] < A[j] then
      Swap A[j] and A[j + 1]
```

$$\sum_{i=1}^{n} \sum_{j=1}^{n-i} 1 \xrightarrow{\text{some math}} O(n^2)$$

6 # SelectionSort

Idea: Swap min into first spot, second-min to second, etc.
(This is hand-wavy on purpose)

85	24	63	45	17	31	96	50
17	24	63	45	85	31	96	50
17	24	63	45	85	31	96	50
17	24	31	45	85	63	96	50 ←
							↑

7 SelectionSort

```
for i ← 1 to n − 1 do
    min ← i
    for j ← i + 1 to n do
        if A[j] < A[min] then
            min ← j
    Swap A[i] and A[min]
```

$O(1)$

$2(n-1) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 \longrightarrow \Theta(n^2)$

8 What's nice about SelectionSort?

- Easy to program
- Easy to explain
- Does it waste memory?
- Does it only work for numbers?
- What other info do we need?
- Are there inputs that are sorted faster?
- Is there a lot of data movement?