CSCI 241
Scott Wehrwein

Sorting algorithm properties:
Stable
In-place
Goals

Know the definition of a stable sorting algorithm.

Know the definition of an in-place sorting algorithm.

Be able to categorize all the sorts we've covered with respect to these properties.
Stability

Objects can be sorted on keys - different objects may have the same value.

A stable sort maintains the order of distinct elements with the same key.

- Example: sort a list of Student objects by first name only
- Example: sorting numbers on 10’s place only
- Example: sort colored numbers

\[ [6 \ 2 \ 6 \ 2 \ 3 \ 4] \]
Stability

A stable sort maintains the order of elements with the same value.

Original: \[6 \ 2 \ 6 \ 2 \ 3 \ 4 \]

Stably sorted: \[2 \ 2 \ 3 \ 4 \ 6 \ 6 \]

Unstably sorted: \[2 \ 2 \ 3 \ 4 \ 6 \ 6 \]
Space Complexity

Time complexity: how many operations?

Space complexity: how much (extra) memory?

- Don’t count the size of the input: we have no choice but to store it!
In-Place

A sort is considered in-place if it requires less than $O(n)$ storage space in addition to the input.
In-Place

A sort is considered in-place if it requires less than O(n) storage space in addition to the input. Example:

```python
insertionSort(A):

    i = 0;
    while i < A.length:
        j = i;
        while j > 0 and A[j] > A[j-1]:
            swap(A[j], A[j-1])
            j--
        i++
```

O(1) space