

CSCI 136:
Data Structures
and
Advanced Programming
Lecture 14
Sorting, part 2

Instructor: Dan Barowy
Williams

Topics

- Bubble sort
- How do we sort data of any type?

Your to-dos

1. Lab 5 (solo lab), **due Tuesday 3/15 by 10pm.**
2. Read **before Fri**: Bailey, Ch 6.5-6.6.

Announcements

1. Midterm review: **Wed, March 16** in class.
2. Midterm: in lab next week, **Thursday, March 17.**
3. No class: **Fri, March 18.**

Study tip

Grades are important, but they are **not the most important thing** in life.



Study tip

A few things to keep in mind.

Quizzes are for **practice**.

Labs are for **practice**.

You can **resubmit** labs.

You can **resubmit** the midterm.

Just do your best.

Sorting algorithms

Sorting algorithm

A **sorting algorithm** is a **procedure** for transforming an unordered set of data into an ordered sequence.

A **comparison sorting algorithm** takes as input a set **S** and a binary relation **<** that defines an **ordering** on **S**.

Bubble sort

6 5 3 1 8 7 2 4

Bubble sort

Bubble sort is an **in-place sorting algorithm** in which the largest element “**bubbles up**” during each pass. Bubble sort makes **$n-1$** passes through the data, performing pairwise comparisons of elements using **<**. During a pass, if bubble sort finds two elements out of order, it **swaps** them.

Bubble sort maintains the **invariant** (an always-true logical rule) that the rightmost **n -numSorted** elements are sorted.

I.e., bubble sort builds a sorted order to the right.

Bubble sort algorithm

```
public static void bubbleSort(int data[], int n)
// pre: 0 <= n <= data.length
// post: values in data[0..n-1] in ascending order
{
    int numSorted = 0;    // number of values in order
    int index;           // general index
    while (numSorted < n)
    {
        // bubble a large element to higher array index
        for (index = 1; index < n-numSorted; index++)
        {
            if (data[index-1] > data[index])
                swap(data, index-1, index);
        }
        // at least one more value in place
        numSorted++;
    }
}
```

Bubble sort complexity

Bubble sort is an **$O(n^2)$** sorting algorithm in the **worst case**. The naive algorithm is also **$O(n^2)$** in the **best case**. With a small modification, bubble sort is **$O(n)$** in the best case (i.e., where the array is already sorted).

Bubble sort's performance is bad enough that there are few practical uses for it (other than for teaching!).

What if...

... you wanted to sort **arbitrary objects**?

What's **problematic** with our bubble sort implementation?

Recap & Next Class

Today:

- More sorting algorithms
- Sorting arbitrary data

Next class:

- Comparators
- More sorts