Algorithmic Art
Day 2

Nina Lutz, MIT

Prepared for Clubes de Ciencias ASU

PICK A COMPUTER WHERE YOU CAN READ THIS! Log in.

Look for processing under downloads/processing. Open a new Processing sketch.

If it is not there download it. Help your neighbors! Raise your hand if you need help. :)

Sorting Algorithms

Used to rearrange elements in a data structure.

Most sorting algorithms rely on a **comparator** between elements. This can be numerical, logical, or alphabetical.

Naive Sorting Algorithm

6 4 2 8 3 1 7

Maybe you start at the first number and for each number you move it to the left or right of the pivot if it’s smaller or larger.

Naive Sorting Algorithm

6 4 2 8 3 1 7

4 **6** 2 8 3 1 7

4 2 **6** 8 3 1 7

4 2 **6** 8 3 1 7

4 2 3 **6** 8 1 7

4 2 3 1 **6** 7 8

4 2 3 1 **6** 8 7 <— not sorted :(:(

Pivots and Comparators

Last example, 6 was our **pivot.** Which is a starting point of a **comparator.**

But rarely can we sort with O(n) comparisons on a stagnant pivot.

We say rarely because we can get lucky. But again, not often.

Considerations of Sorting Algorithms

Speed - How fast it is

Memory - How much space it takes up

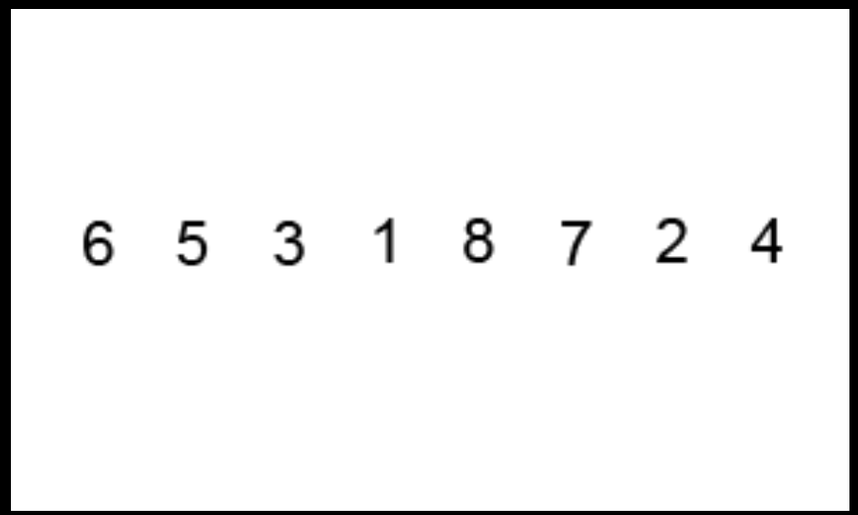
Stability - sorting repeated elements in the same order as original structure

Comparator - Most comparisons are constant time but some aren’t

Bubble Sort

Swaps by each pair.

Then goes through the new pairs until the list is sorted.



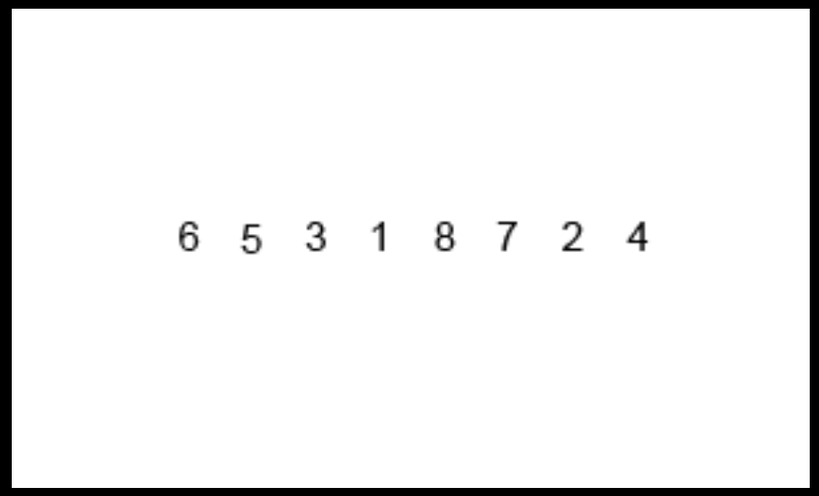
O(n^2), best case
O(n)

Merge Sort

Divide and conquer approach to sort.

Divides array into sub arrays and sorts from smallest size to largest.

O(nlgn) worst case, but in many cases can be O(n).



Activity

Make a drawing about merge or bubble sort. Make it creative. Tell a story. Or make it visual. Relate it to art you like.

It doesn’t have to be perfect!

10 min — then we’ll come back and talk them out.

Ok let’s do some code

Bubble Sort PseudoCode

For each pair in array:

if left > right: SWAP

if no swaps occurred: DONE

if swaps occurred: REPEAT

Let’s try to code it!