Agenda

Intros:

WWCode @ Alchemy Code Lab

Link to slides:
https://github.com/wwcodeportland/study-nights/tree/master/algorithms

Topic Summary:

Sorting algorithms - two common options

Lab Time:

Implementing the sorting algorithms

Algorithms Study Night

Leadership Team

Caterina
Director

Richa
Director

Sarah Joy
JavaScript Lead

Shiyuan
Design Lead

Tricia
DevOps Lead

Keeley
Open Source Lead

Alia
Algorithms Lead

Keeley
Networking Nights Team

Posey
Community Engagement Manager

Upcoming Events - October

- DevOps Study Night: What's in the Box!? @ Vevo
  - Wednesday, October 4th, 6 PM
- "4k 4Charity" Fun Run @ Tom McCall Waterfront Park
  - Thursday, October 12, 3 PM
- Design + Product Study Night - Introduction to Sketch Workshop @ New Relic
  - Tuesday, October 17, 5:30 PM
- Networking Night @ AKQA
  - Wednesday, October 18, 5:30 PM
- JavaScript Study Night @ Metal Toad
  - Wednesday, October 25, 5:30 PM

{short} Code of Conduct

Women Who Code (WWCode) is dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, or creed. Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. Because we value the safety and security of our members and strive to have an inclusive community, we do not tolerate harassment of members or event participants in any form. Our **Code of Conduct** applies to all events run by Women Who Code, Inc. If you would like to report an incident or contact our leadership team, please submit an **incident report form**.

Resources

WWCode @ Meetup.com

WWCode @ Slack

WWCode @ Github

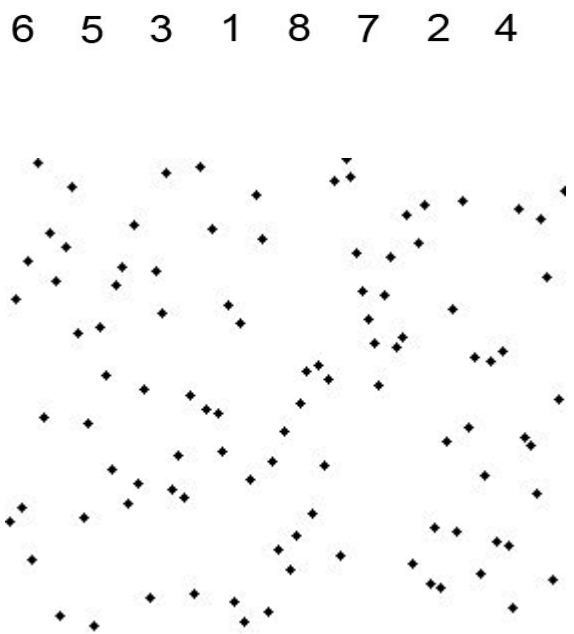
Big-O CheatSheet

Bubble Sort Algorithm

To implement bubble sort:

1. For each value, compare it to the one next to it, and swap if the right is lower than the left.
2. Move through the list one by one, repeating step one.
3. Do this repeatedly until there are no changes throughout the whole list.

Average and worst case Big(O) complexity for this algorithm are $O(n^2)$.



Bubble Sort Example

5 3 1 4

1. 5 3 1 4
2. **3 5** 1 4
3. 3 **1 5** 4
4. 3 1 **4 5**
5. **1 3** 4 5
6. 1 **3 4** 5
7. 1 3 **4 5**
8. **1 3 4 5**

1. Original set to be sorted
2. Look at the two first elements and swap if they are not already in the right order
3. Look at the 2nd and 3rd element and swap if they are not already in the right order
4. Repeat until you've reached the end of the set
5. Go back to the beginning of the set and repeat
8. If no swap happened and you've reached the end of the set, you're done!

Merge Sort Algorithm

Conceptually, a merge sort works as follows:

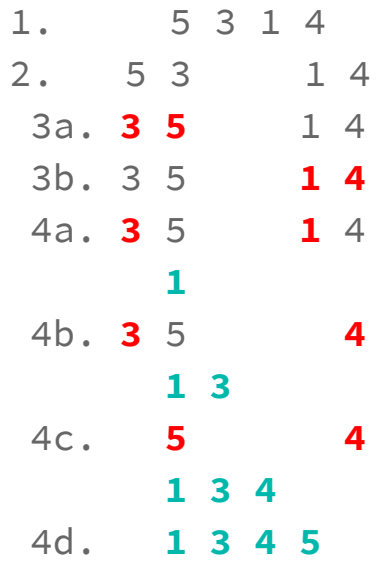
1. Divide the array in half
2. Divide the halves by half until 2 or 3 elements are remaining
3. Sort each of these halves
4. Merge them back together

Average and worst case Big(O) complexity for this algorithm are O(n(log(n))).

6 5 3 1 8 7 2 4

Merge Sort Example

5 3 1 4

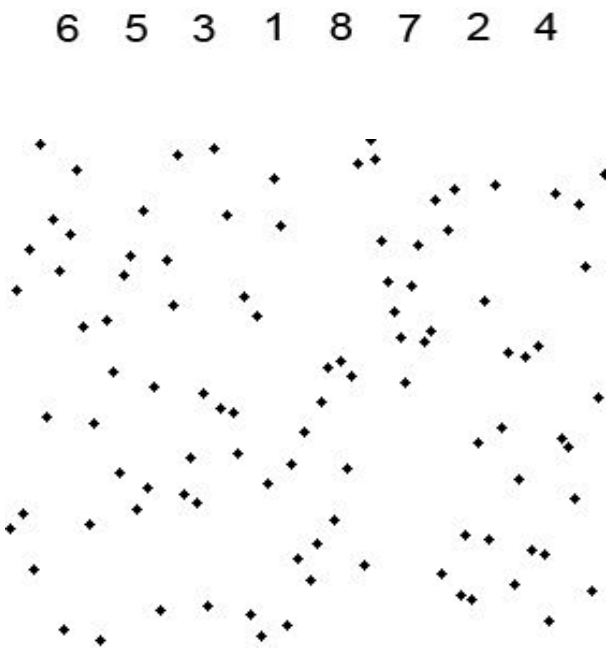


1. Original set to be sorted
2. Separate into sets of two
3.
    a. Compare 5, 3 and pick lower number
    b. Compare 1, 4 and pick lower number
4.
    a. Compare 3, 1 and pick lower number
    b. Compare 3, 4 and pick lower number
    c. Compare 5, 4 and pick lower number
    d. Pick last number

Quick Sort Algorithm

Steps for QuickSort:

1. Pick a “pivot” element.
2. “Partition” the array into 3 parts:
   a. First part: all elements in this part is less than the pivot.
   b. Second part: the pivot itself (only one element!)
   c. Third part: all elements in this part is greater than or equal to the pivot.
3. Then, apply the quicksort algorithm to the first and the third part. (recursively)



Average Big(O) complexity is O(n(log(n))) while worst case is $O(n^2)$.

QuickSort Example

5 3 1 4

1. 5 **3** 1 4
2. **5** **3** 1 **4**
3. **5** **3** **1** 4
4. **1** **3** 5 **4**
5. **1** **3** **5** **4**
6. **1** **3** **4** **5**

1. Select your pivot (3)
2. Select the first value to the left of the pivot, and the last value to the right of the pivot and compare. Swap if right is less than pivot.
3. Move up one spot on the right side, repeat step 2.
4. Left side is now lower than pivot. It is done, as is pivot. Pick new pivot (4)
5. Repeat steps for right side of original pivot until everything is sorted.

3 Sorting Algorithms solutions

1	Merge sort	2	Quicksort	3	Other sorting algorithms
• Wikipedia explanation/pseudocode/c-like solution • Java • Python • Javascript		• Pseudocode • Java • Python • Javascript		• Wikipedia	