

Agenda

Intros:

WWCode @ Code Fellows PDX

Link to slides:

<https://github.com/wwcodeportland/study-nights/tree/master/algorithms>

Topic Summary:

An Approach to Solving Algorithms

Lab Time:

Pair Programming + 3 Algorithms

Algorithms Study Night



Code Fellows PDX

Leadership Team



Caterina
Director



Richa
Skills Development Lead



Shiyuan
Design Lead



Tricia
DevOps Lead



Sabina
Events Lead



Sarah Joy
JavaScript Lead



Keeley
Community Lead



Alia
Algorithms Lead

Upcoming Events - July

Rest of July:

- [Design + Product Study Night @ New Relic](#) - Tue, July 18th, 5:30 PM
- [JavaScript Study Night @ Metal Toad](#) - Wed, June 26th, 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](#)

How to Approach Solving Algorithms

1. DON'T start with your computer !!!
2. Write down **inputs/outputs**.
Make sure you know all of the information about the problem you are solving.
3. Try a single example **by hand** to see expected results.
4. Think about any possible **edge cases**.
5. **High level solution** - pseudocode. Optimize if you can

Continued...

6. Choose your **data structure**.

How often will you be accessing? How often will you be inserting/deleting?
Data Structure complexity.

7. **Code** your solution

8. Discuss the algorithm **complexity**

9. **Optimize** if possible

Let's try that together

Sample Problem

Write a function that tells whether or not a given number is part of a listing. Numbers in that listing are sorted.

Step-by step example

1. DON'T start with your computer !!!



Step-by step example

2. Write down inputs/outputs.

“Write a function that tells whether or not a given number is part of a listing of numbers. Numbers in that listing are sorted.”

Inputs: array of integers + integer

Output: boolean

Step-by step example

3. Try a single example by hand to see expected results.

If input is

1 4 6 17 46 78 79 178 77777 , 178

Output is

true

Step-by step example

4. Think about any possible edge case.

Main concerns depending on language:

- `null/undefined` inputs
- `empty` inputs
- `overflow` (ex: number getting higher than `MAX_VALUE`)
- if types are defined and don't represent the `meaning` of the data:
 - String for integer
 - Object for Integer



Step-by step example

5. High level solution - pseudocode.

“Write a function that tells whether or not a given number is part of a listing of numbers. Numbers in that listing are sorted.”

Brute force:

- read each number in the array
- compare with the number you are trying to find



Step-by step example

5. High level solution - pseudocode.

“Write a function that tells whether or not a given number is part of a listing of numbers. Numbers in that listing are sorted.”

Brute force:

```
read each number in the array smaller than the number to find
    compare
```

Step-by step example

6. Choose your data structure.

How often will you be accessing? How often will you be inserting/deleting?
Data Structure complexity.

Pseudo-code :

```
    read each number in the array smaller than the number to find
    compare
```

No need to manipulate the data here

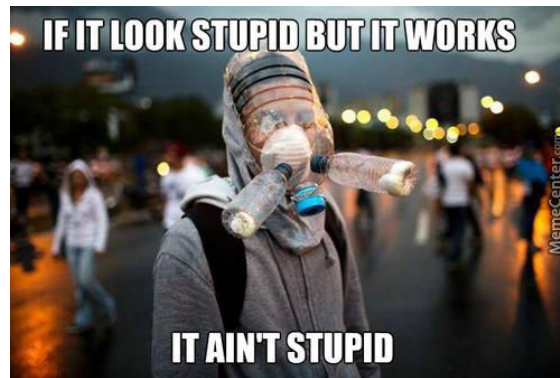
Step-by step example

7. Code your solution

Pseudo-code:

read each number in the array smaller than the number to find
compare

name of function	<code>boolean findNumber(int[] listing, int number){</code>
name of input	<code>for(int i=0; i<listing.length && listing[i]<=number; i++){</code>
name of variable	<code>if(listing[i] == number) return true;</code>
for or while loop ?	<code>}</code>
	<code>return false;</code>
	<code>}</code>



Step-by step example

8. Discuss the algorithm complexity for loop => $O(n)$

9. Optimize if possible

What if instead of reading all the numbers we only looked at the first last and middle numbers ?

1 4 6 17 46 78 79 178 77777, 178

46 78 79 178 77777

79 178 77777

“Divide and conquer”

Big gain for big arrays. Here 3 iterations instead of 6

Step-by step example

6. Code your solution

Pseudo-code:

- at each iteration

- look at the middle and pick a side

Step-by step example

7. Code your solution

Pseudo-code:

- at each iteration
 - look at the middle and pick a side

```
boolean findNumber(int[] l, int nb){  
    if(l.length == 0) return false;  
    int minInd = 0;  
    int maxInd = l.length-1;  
    if(nb==l[minInd] || nb==l[maxInd]) return true;  
    while (minInd != maxInd){  
        int midInd = minInd + maxInd-minInd/2;  
        if(nb==l[midInd])return true;  
        if(l[midInd] > nb) maxInd = midInd;  
        else minInd = midInd;  
    }  
    return false;  
}
```

Step-by step example

8. Discuss the algorithm complexity for loop => $O(\log(n))$
9. Optimize if possible

Practice

1 | The Triforce

- [Problem Statement](#)

2 | French License Plates

- [Problem Statement](#)

3 | Coding Game

- [CodinGame](#)