Things you should know!

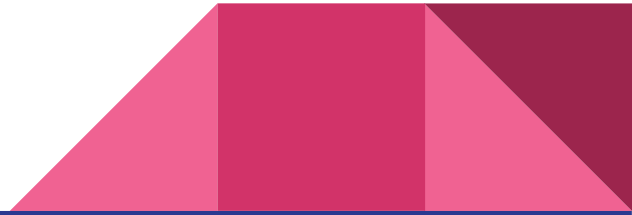
- Pick up an ICI sheet!
- Today’s attendance is... a Sign-In Sheet!
- Check the Slack!!!

Week 4!

CMSC389O: The Coding Interview

Today

- Arrays, arrays, arrays...
- In Class Interviews (ICI)



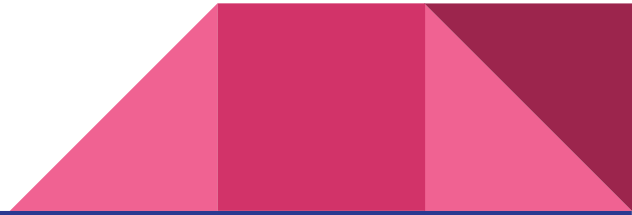
Arrays

- One of the most *important* topics for programming interviews
- Most companies ask array questions at some point
- String problems can be structured like Arrays

- Usually the first data structure people learn about
- Has specific upper and lower bounds (i.e. *IndexOutOfBoundsException*)
- Can be iterated through very easily
- Has lots of neat tricks to make your life easier, if you look closely!

Types of Problems

- Properties of Arrays (Does this array contain X? How do you check if…)
- 2D Matrices, *N*D Arrays
- ArrayList (“endless” array; can be added to)
- Character Arrays (Strings)
- Missing Elements
- Sorting & Searching
- Serializing
- Efficiency

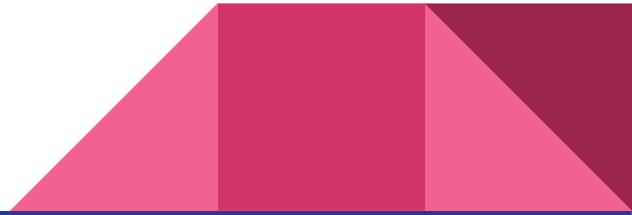


Things to Think About

- Is this array already sorted?
- Does this array have any negative values? Does that affect my solution?
- Can I sort it, and still have the most efficient solution? (Better than *O(nlogn)*?)
- Did I check my upper and lower bounds?
- Would traversing in reverse make it more efficient?
- Can I keep track of values in O(1) space? O(n) space?
- Edge cases!

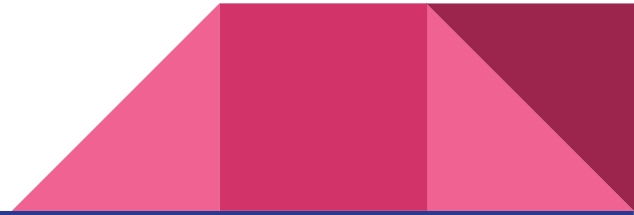
Edge Cases

- Empty sequence
- Sequence with 1 or 2 elements
- Sequence with repeated elements
- If you encounter elements that are not allowed
- What to return in these cases



Types of Solutions

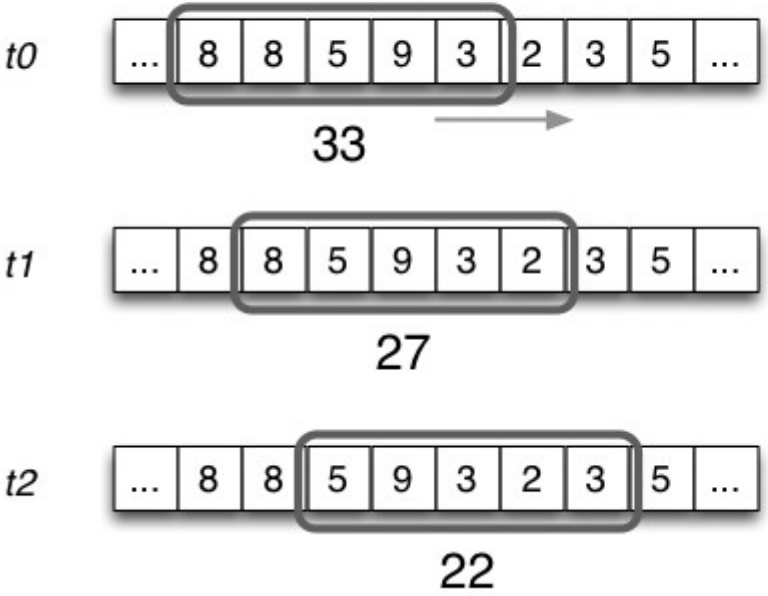
- Sliding Window Technique
- Fast *i*, slow *j*
- Mergesort, Quicksort, Heapsort (*O(nlogn)* sorts)
- Use Array Elements as Index (Bijection from $\mathbb{N} \rightarrow$ Elements)
- Straight run through
- Looping



Sliding Window

- Used to get a contiguous subsequenc of an array
- Questions like:

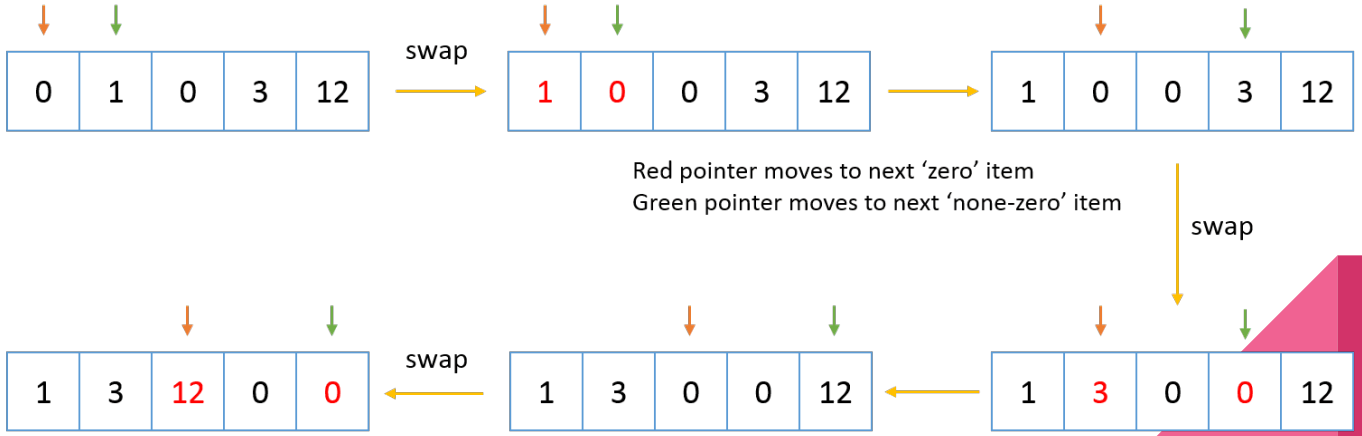
  “Given an array of size n and a number k, find the minimum summat of k elements.”



Fast i, Slow j

- Variation of *Sliding Window*
- Has a *reaching i*, and a *stationary j*
- Good for in-place modification of an array

Red pointer points ‘Zero’ item
Green pointer points ‘None-zero’ item



Mergesort, Quicksort, Heapsort

- Will discuss next week!
- Important parts of each:

Mergesort	*Quicksort*	*Heapsort* ( )
- Merge action - For huge datasets	- Partition - Close to Binary Search, with specific size on each side	- Heapify - Can use MinHeap or MaxHeap to get *k* number of Min/Max elements of array

Use Array Elements as Index

- Create a new array (*O(k) extra space*), and store TRUE (1) or FALSE (0) in the indices whose values correspond with the element you are keeping track of
- OR use your original array (*O(1) extra space*) to mark presence of an element x by changing the value at the index x to negative
  - Acts like a flag!
  - Start array at 1 for easier application
  - Check out *Counting Sort*! :-)

	1	2	3	4	5	6
A	7	3	8	1	2	5

	1	2	3	4	5	6	7	8
A2	1	1	1	0	1	0	1	1

	1	2	3	4	5	6
A3	−7	−3	−8	1	−2	5

In-Class Interviews

- Match up with someone who has a DIFFERENT question than you!
  - (Hint: There are only two questions to be asked…)

Reminders

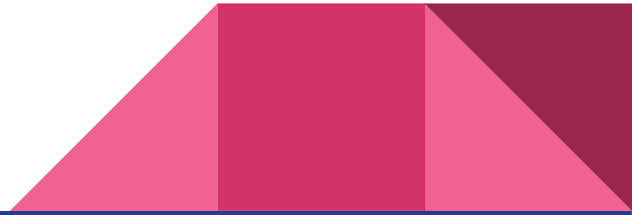
- Fill out feedback form at http://ter.ps/f9h !
- Career Fair!!!
  - February 20, 2018
  - 4PM - 8PM
  - Seniors start at 3PM
  - COLLEGE PARK MARRIOTT HOTEL & CONFERENCE CENTER
  - https://cs.umd.edu/cscareerfair/students for more information!

HW1 From Last Week...

- We’ve updated GitHub with the correct Space and Time Complexities
- Scale:
  - -2 for less than O(n) time complexity for part 1
    - Common mistake: sorting array
  - -2 for less than O(1) space complexity for part 1
    - Common mistake: using Hashset or other data structure
  - -2 for less than O(n) time complexity for part 2
    - Brute force solution
  - -2 for less than O(1) space complexity for part 2

Homework Due for Next Week

https://github.com/UMD-CS-STICs/389Ospring18/blob/master/HW3_Arrays.md



Feedback:

- Pros
  - Positive feedback about HW
  - Positive feedback about ICIs
- Cons
  - Clerical errors
- Things to change
  - Adding time/space complexity to HWs
  - More time for ICIs