

Things you should know!

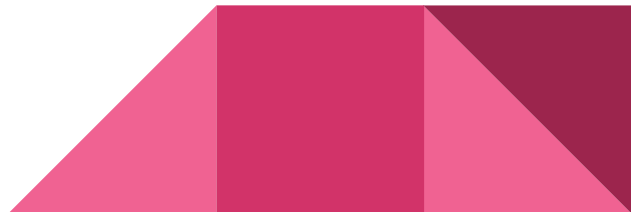
- Pick up an ICI sheet!
- Today's attendance is... a Sign-In Sheet!
- Use the Slack if you are struggling with concepts and to ask about previous weeks' homeworks!!!
- We have office hours by appointment!
 - We can do mock interviews with you
 - Message us for tips if you have an interview coming up
- Didn't find your dream job at the career fair?
 - Check out these research opportunities
 - https://www.nsf.gov/crssprgm/reu/reu_search.jsp
- Midterms are the week of March 12th (3 weeks away)
 - Will release time slot signups on March 2nd

Week 5!

CMSC3890: The Coding Interview

Today

- Searching and sorting
- In Class Interviews (ICI)



$O(n^2)$ sorts-

- Bubble sort

- bubble largest values to the end

6 5 3 1 8 7 2 4

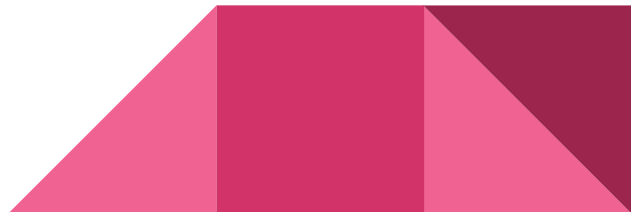
- Insertion sort

- go through the array, putting each element in the correct place relative to other already sorted elements (requires shifting all subsequent

6 5 3 1 8 7 2 4

- Selection sort

- pick smallest element and move to the front; recurse on unsorted elements



$O(n \log n)$ sorts-

● Quicksort

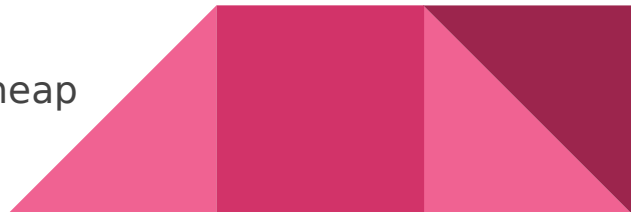
- Continually divides the set by the average, until the set is recursively sorted.
- Best Case Sort: $O(n)$
- Average Case Sort: $O(n \log n)$
- Worst Case Sort: $O(n^2)$

● Mergesort

- Divides the set into the smallest possible groups immediately then reconstructs the incrementally as it sorts the groupings.
- Best Case Sort: $O(n)$
- Average Case Sort: $O(n \log n)$
- Worst Case Sort: $O(n \log n)$

● Heapsort

- Put elements into a heap and continually pop off the heap



Note on Heapsort

Use PriorityQueues to implement heaps (don't use a minheap)

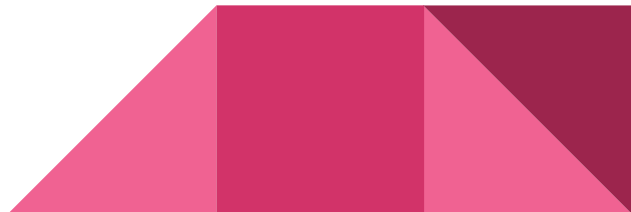
** know your language!!



Quicksort

1. Pick pivot
2. Partition
3. Recurse on both halves

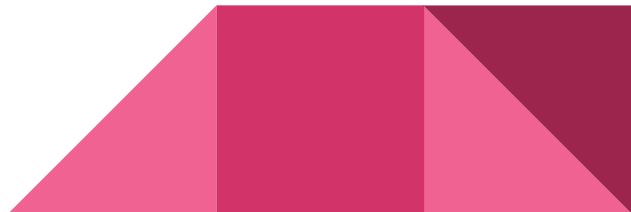
- Worst case scenario?
- Best case scenario?



$O(n)$ sorts-

- Radix sort
- Counting sort
- Bucket sort

Why not always use linear time sorts??



Example Question

Given two strings `s` and `t`, write a function to determine if `t` is an anagram of `s`.

For example,

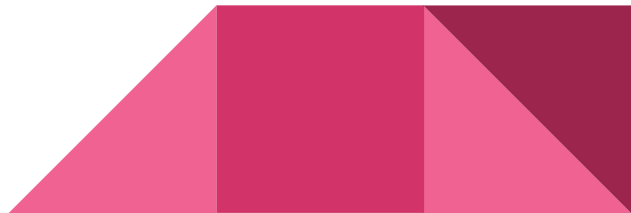
`s = "anagram", t = "nagaram"`, return `true`.

`s = "rat", t = "car"`, return `false`.



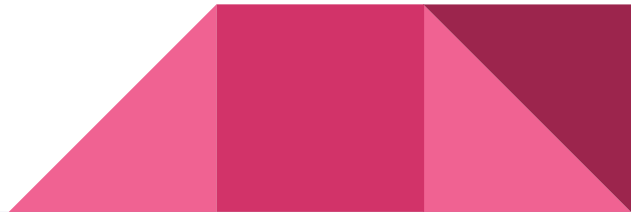
Searching

- Linear Search
- Binary Search
- Depth First Search / Breadth First Search



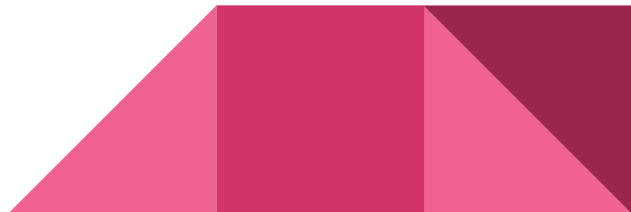
When should I use this stuff?

- Binary search
 - Validation
- $O(n)$ sorts
 - numbers/letters have small range
- $O(n \log n)$ sorts
 - these are your standard sorting algorithms
- $O(n^2)$ sorts
 - know these conceptually, but use faster sorts whenever possible



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 <https://goo.gl/forms/5135cM8oF8AAmMLN2>!
- Send us your photos from the career fair for extra credit!



Homework Due for Next Week

https://github.com/UMD-CS-STICs/3890spring18/blob/master/Week5/HW4_SandS.md

