2020-04-21 Sorting

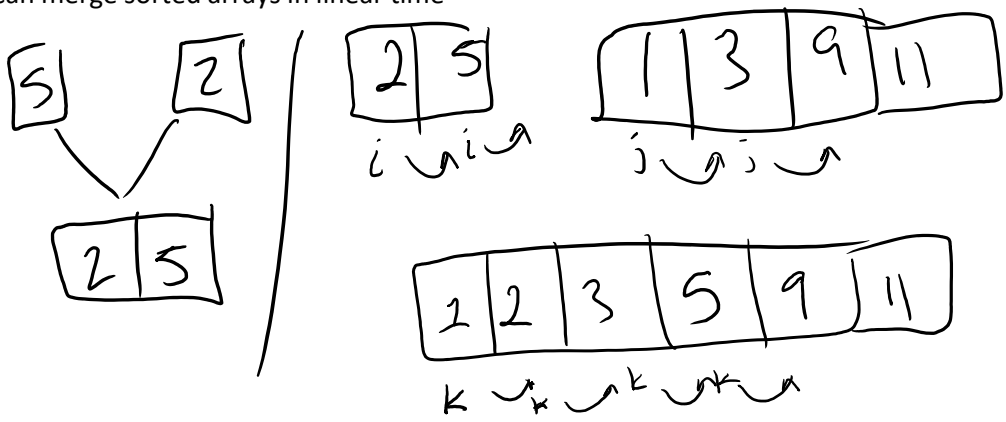
Tuesday, April 21, 2020 10:43 AM

Final Project

- Any idea is okay, as long as it's in C++ or C#
- Focus should be on delivering a cool project / product
  - Tech stack suggestion: Unity (C#, games), ASP.NET MVC (C#, websites), QT (C#/C++, traditional desktop apps), WPF (C#, traditional desktop apps), WinForms (C#/C++, traditional desktop apps)
- By tomorrow evening, you must:
  - Create a new github repo
  - Create a readme in that repo that outlines your project
  - Submit link to repo on canvas
- Code checkin next Thurs/Friday

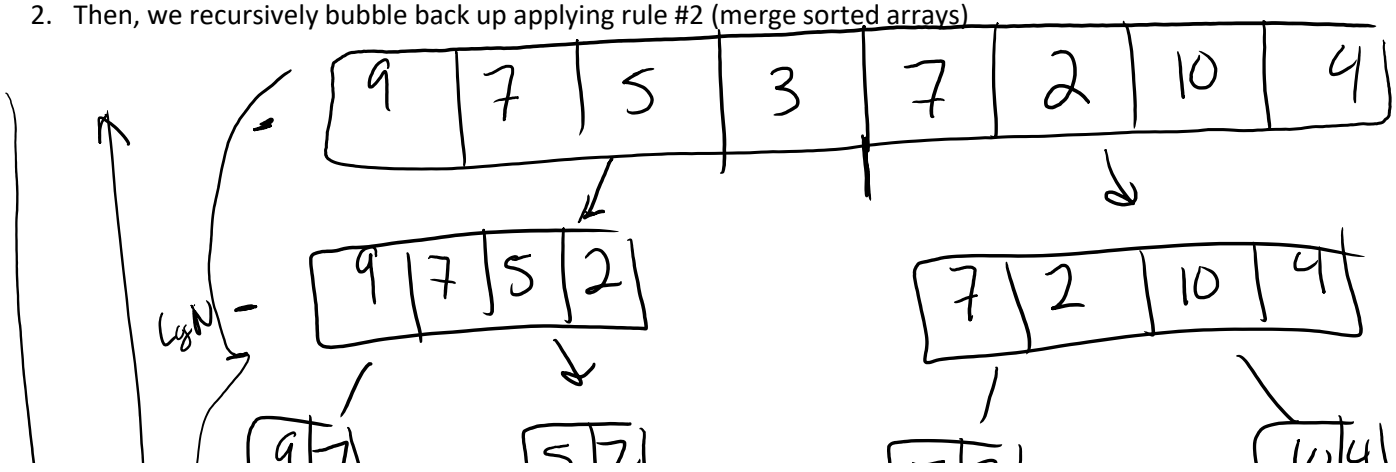
Student Observations

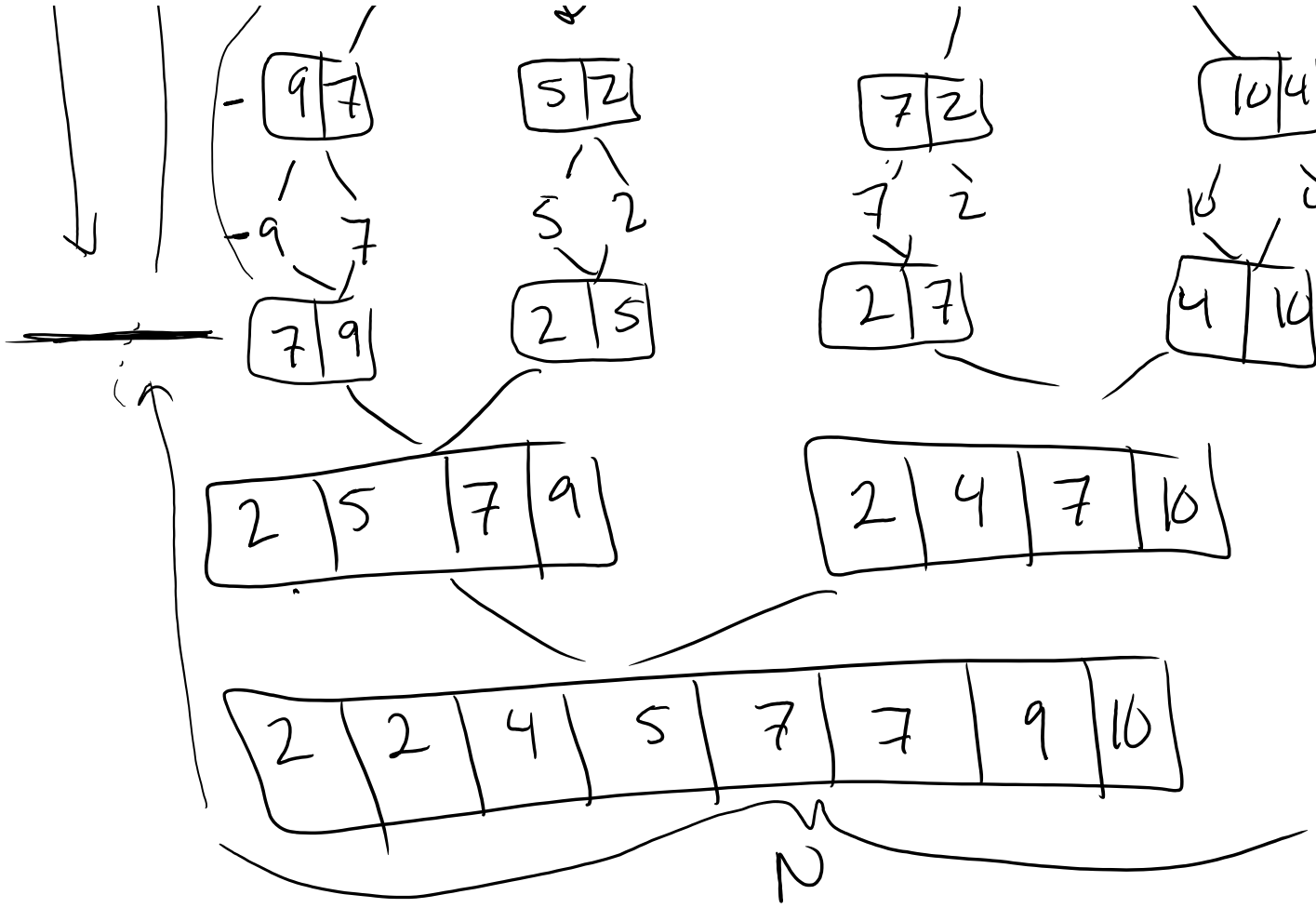
- Balancing time vs space complexity
  - Really depends on "functional requirements"
- Merge sort: III
  - Runtime analysis
  - Relies on two fundamental observations:
    1. An array of size 1 is already sorted
    2. You can merge sorted arrays in linear time



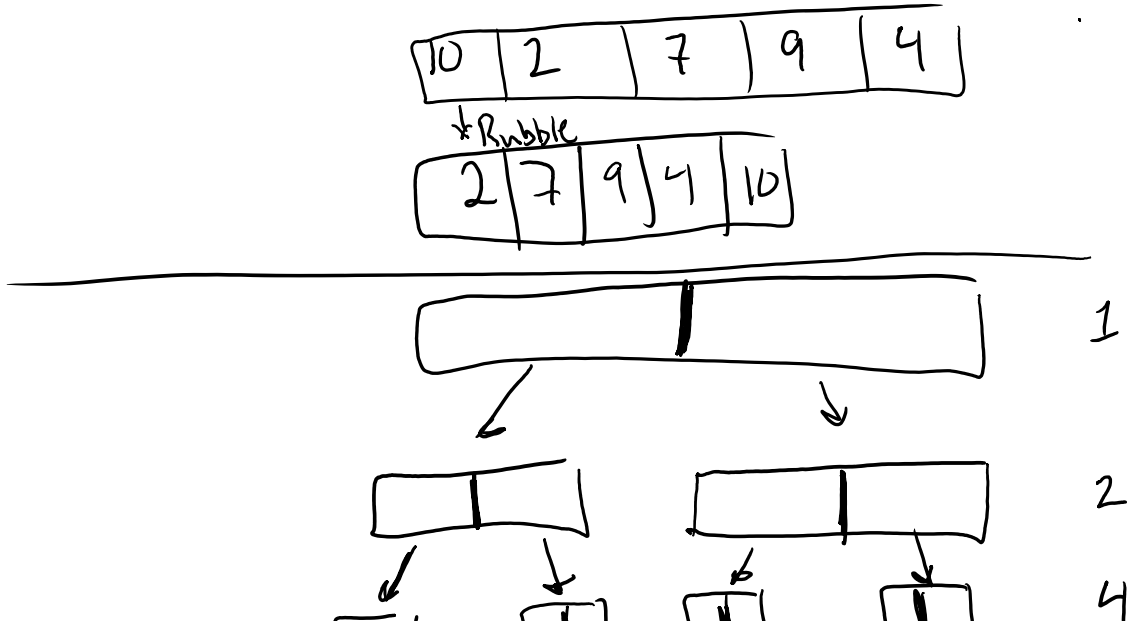
General Mergesort Algorithm

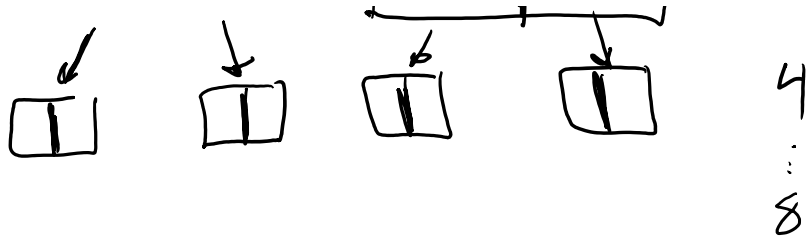
1. Recursively break down a large array until we get to observation #1 (arrays of size 0)
2. Then, we recursively bubble back up applying rule #2 (merge sorted arrays)





- Shell Sort: IIII
  - Gap values
  - Differentiator between n^2 and an NLogN sort is that at the end of each iteration an n^2 sort will only have moved at most a single value into its appropriate location. NLogN sorts move multiple things into their appropriate location per iteration.





- Shell sort is insertion sort that creates subarrays out of N-spaced elements. As shell sort progresses, N shrinks by some amount (thereby creating less sub-arrays) until N reaches 1. At which point we run normal insertion sort.

- Radix Sort: II
  - Runtime analysis
  - Applying with other data

Sort {12, 7, 99, 38, 404}

time →

	1	2	3
0		404, 7	7, 12, 38, 99
1		12	
2	12		
3		38	
4	404		404
5			
6			
7	7		
8	38		
9	99	99	

$O(N \cdot K)$? $O(N \log N)$

$K = 10^k = \text{largest}(\text{number})$