

Mock Interview Debrief

CSE 332 Summer 2020

Instructor: Richard Jiang

Teaching Assistants:

Hamsa Shankar Kristin Li Winston Jodjana

Maggie Jiang Hans Zhang Michael Duan

Jeffery Tian Annie Mao

Lecture Q&A: pollev.com/332summer

Lecture clarifications: tinyurl.com/332-08-21A

Lecture Outline

❖ **Mock Interview Debrief**

- KMerge
- General Guidance
- Tips for Improving

❖ Common Interview Question Patterns

❖ Staff Q&A!

pollev.com/332summer :: tinyurl.com/332-08-21A

KMerge: Problem Statement

- ❖ You are given k sorted arrays, each with n numbers in them. Give an algorithm for merging these arrays into a single sorted array of numbers.
- ❖ Time Complexity Constraint of $O(nk\log(k))$.
- ❖ Input:

```
[ [1, 3, 3, 6, 7],  
  [2, 4, 5, 6, 10],  
  [1, 6, 8, 9, 9],  
  [3, 5, 7, 8, 11] ]
```
- ❖ Output:

```
[1, 1, 2, 3, 3, 3, 4, 5, 5, 6, 6, 6, 7, 7, 8, 8, 9, 9, 10, 11]
```

KMerge: Brute Force Solutions

- ❖ **Brute Force 1:** Move everything into an array, run your favorite sorting algorithm on it. Total # of elements is $n \cdot k$

- Runtime? " $n^2 \log n$ " $\rightarrow nk \log(nk)$

- ❖ **Brute Force 2:** Move everything into an heap, remove elements from the heap one by one and put it into a resulting array.

- Runtime? $nk \log nk$
heap build

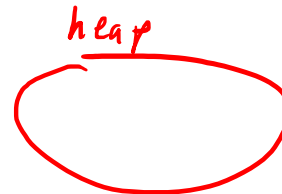
BuildHeap version: $nk + nk \log nk$

KMerge: More Sophisticated Solutions

- ❖ **The Heap Way:** Create a heap and only keep k elements in it, one from each subarray. Remove an element, put it in the result array, and replace it in the heap with the next element from the same subarray.

- Why can we only keep k elements in the heap? *new class or hash table*
- Why do we need to keep track of where elements come from?
- **Runtime?** $K + NK \log k$

[1, 2] [3, 4] [2, 3]



logk insert & delete

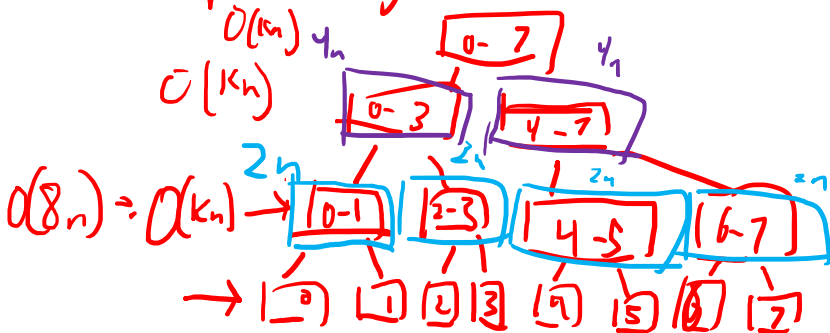
[1, 2, 2, 3, 3, 4]

KMerge: More Sophisticated Solutions

- ❖ **The Merge Way:** Divide and conquer the subarrays, once you reach the base case of 1 subarray, merge the arrays on the way up.

- What is the merge runtime? $merge(a, b) \rightarrow O(a+b)$
- How much work per level? $O(kn)$ work
- How tall is the recursive tree? $\log k$ height
- Runtime? $O(kn \log k)$

8 input arrays



KMerge: Takeaways and General Guidance

- ❖ **Always begin with clarifying questions:**
 - About edge cases, input, output, clarify your interpretation
- ❖ **Explain your approach before coding:**
 - Explain the brute force way first
 - Explore different data structures, talk about trade-offs
 - Plan with psudeocode/whiteboarding
 - Figure out the runtime before starting
- ❖ **Keep talking while coding:**
 - Relate what you are coding back to your pseudocode
 - If you make a change/optimization, explain why, flex your good coding skills

KMerge: Takeaways and General Guidance

❖ After you finish...

- Run through your code with sample input
- Talk about some tests you might write for the code

❖ Keep it a conversation with your interviewer

- Don't get stuck up on syntax, ask your interviewer if they know or if you can mock it
- If you get stuck, you can always ask your interviewer for a directed hint

❖ When approaching problems, try to notice the special properties of the problem

Practice Tips

- ❖ Review common interview question types/patterns
- ❖ Practice writing pseudocode and explaining runtime
- ❖ Talk to yourself and explain while you code
- ❖ Record yourself and watch it back to see where you can improve
- ❖ Keep mock interviewing! Ask your friends to mutually interview each other.

Lecture Outline

- ❖ Mock Interview Debrief
 - KMerge
 - General Guidance
 - Tips for Improving
- ❖ **Common Interview Question Patterns**
- ❖ Staff Q&A!

pollev.com/332summer :: tinyurl.com/332-08-21A

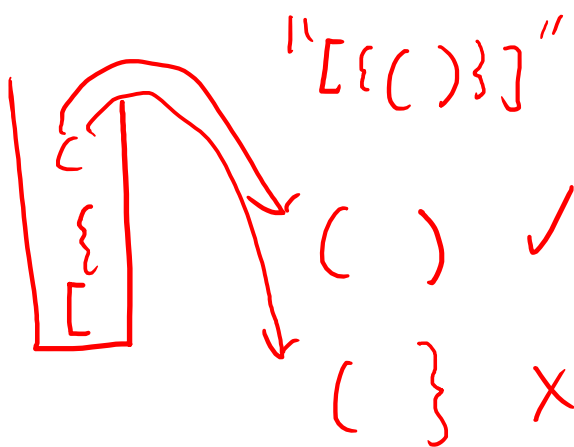
Valid Parentheses

- ❖ Given a string containing just the characters '(', ')', '{', '}', '[', and ']', determine if the input string is valid.
- ❖ An input string is valid if:
 - Open brackets must be closed by the same type of brackets.
 - Open brackets must be closed in the correct order. **[()]**
 - Note that an empty string is also considered valid.
- ❖ Any useful data structures?

Special property - match the most recent bracket

Valid Parentheses

- Given a string containing just the characters '(', ')', '{', '}', '[', and ']', determine if the input string is valid.



$n = \text{string.length}$

$O(n)$

Maximum Subarray Sum

- ❖ Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

- ❖ **Input:** [-2, 1, -3, 4, -1, 2, 1, -5, 4]

- ❖ **Output:** 6 → [4, -1, 2, 1]

- ❖ What's the brute force way? *check every contiguous subarray $O(n^2)$*
- ❖ Can we make it $O(n)$?

sliding window

Maximum Subarray Sum

❖ **Input:** [-2, 1, -3, 4, -1, 2, 1, -5, 4]

❖ **Output:** 6 \rightarrow [4, -1, 2, 1]

max = 6

sum = 5

special property: contiguous

$O(n)$

Alien Dictionary

- ❖ An alien gives you a page of their dictionary and tells you that it is in **alphabetical order** for their language. Can you figure out the ordering of their letters?
- ❖ **Input:** Rar -> Rah -> Raah -> Tuh -> Tttr -> Thr -> Thu
- ❖ **Output:** r -> u -> t -> h -> a *relationship between letters*
- ❖ What's the special property of the input? *alphabetically sorted*
- ❖ What data structures might be useful?
Graph

Alien Dictionary

- ❖ **Input:** Rar -> Rah -> Raah -> Tuh -> Tttr -> Thr -> Thu
- ❖ **Output:** r -> u -> t -> h -> a

r → h

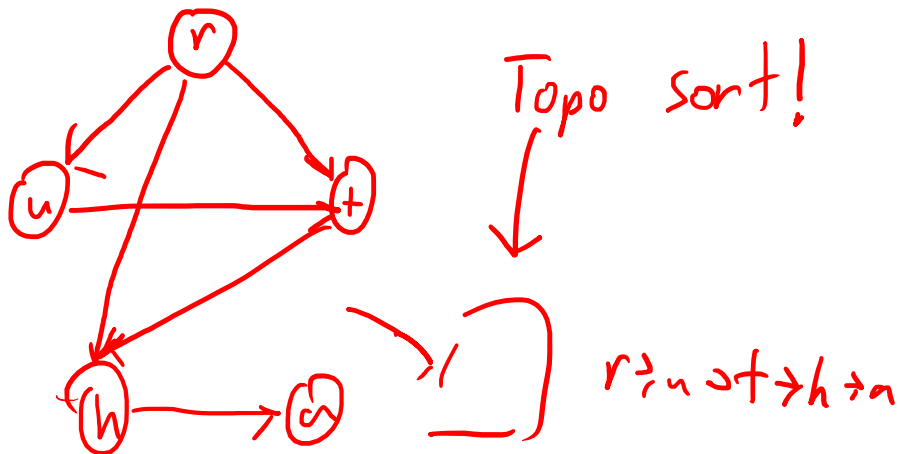
h → a

r → t

u → t

t → h

r → u



Some other types of questions

❖ **Class Design:**

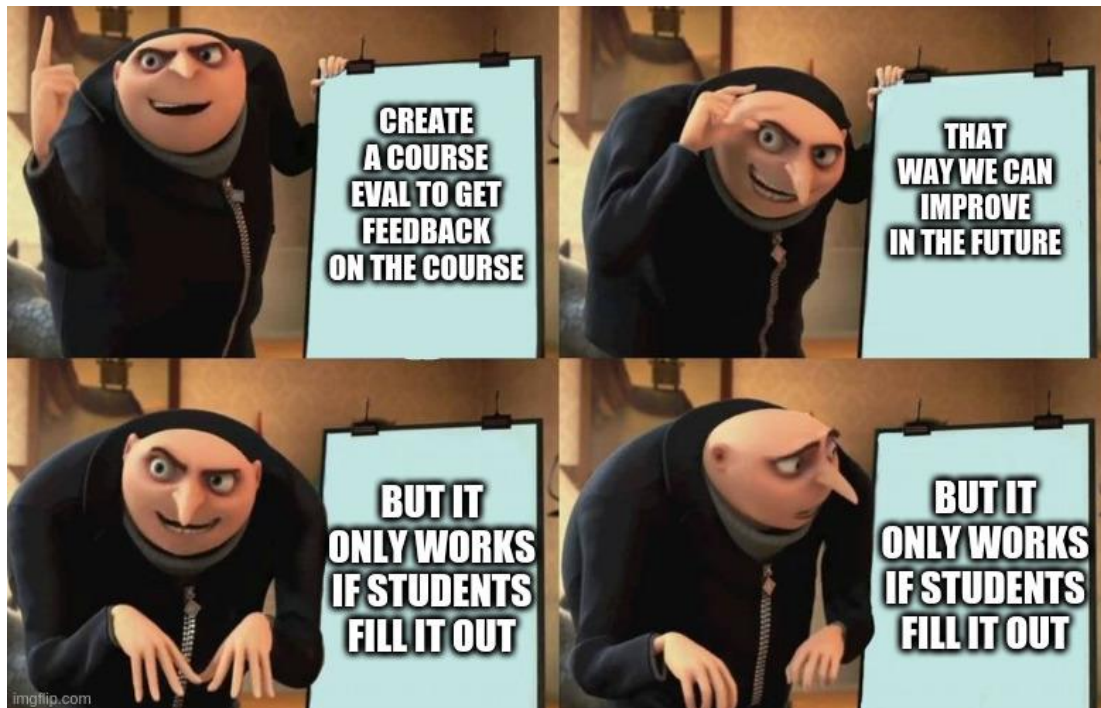
- Can you design a deck that represent a deck of cards? Or schedules meeting rooms?
- Very important for these types of questions to ask clarifying questions.
- Common twist – Concurrency!

❖ **Dynamic Programming:**

- Way to complicated to dive into now
- Essentially, save partial answers to save time
- This is taught in Algos, CSE 421

Course Evals!

<https://uw.iasystem.org/survey/228732>



You've made it!

Staff Q&A Time



Please don't be shy in the halls, come say hi!



Ask us anything!

