Implementation

ACM Club Week 3

Implementation

- “Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?” - *Brian Kernighan*
  - This is bullcrap
- Clever ideas are the ones that make your code simpler
- Listen to what your teachers have told you
  - Don’t micro-optimize (mega-optimizing is ok though… usually)
  - Use descriptive names
  - Comments are good for hard things

Overview

- Recursion < Iteration (Generally)
- Skipping the Simulation
- Sorting
  - Lexicographically minimal string concatenation

Recursion vs Iteration

- Recursion is super cool but
  - Hard to analyze and code
  - Almost impossible to debug
  - Slower
- Everything that can be done recursively can be done iteratively
  - But translating between the two can be hard and pointless.
- When to use Recursion
  - Can’t find a iterative solution
  - Iterative solution is too hard to understand
    - Many problems are presented recursively to students as recursion is easier to think about
    - Example: Petr and Combination Lock

Skipping the simulation

- Sometimes you don’t need to solve the problem to get the answer.
    - Example: New Year and The Treasure Geolocation
    - Example: Maximum number of 2x1 domino's you can place on a n by m grid
- Makes problems far easier but requires cleverness.
- Worth thinking about when you only need to find only the result given by the optimal strategy, but not the optimal strategy

Sorting

- There are many sorting algorithms (Heap sort, Quick sort, Bubble sort, …)
    - If you ever write one in a coding interview, you will be rejected
- Know how to use your language’s sort function
- Sometimes it can be less obvious than sorting an array of integers
- Example: Find the lexicographically minimal concatenation of some numbers

Lexicographically Minimal Concatenation

Concatenate a set of strings to form the lexicographically minimal (alphabetically first) string.

Examples:·

{a, ab, abc} ↣ [a][ab][abc]

{a, ba, ab} ↣ [a][ab][ba]

If you can’t solve this in an hour, then you need to “stop lying to yourself … stop calling yourself a ‘Software Engineer’”

Lexicographically Minimal Concatenation

- What doesn’t work: sorting the array based on what strings are lexicographically minimal and then concatenating (counterexample: {c,cb})
- How can we salvage the idea?
- Make a custom comparator and sort using that
    - How can we determine which way is better to concatenate two strings?
    - Just do it! For two strings str1 and str2, compare $\overline{\text{str1str2}}$ with $\overline{\text{str2str1}}$

Problems for the week

https://github.com/OSUACM/Weekly_Events

Next week: Lecture by Professor Kenneth Supowit