South Pacific ICPC Divisionals 2018 Editorial

January 9, 2019

Problem E : Easiest Problem (Easy)

- Ordering by average is equivalent to sorting by sum
- Compare by sum then by captain's ranking to break ties
- Keep running best name and difficulty scores
- $O(P)$
- $O(P\log(P))$ also runs in time

Problem F : Flow (Easy)

- Flavour text is deliberately obfuscatory
- There is a strategy for optimally assigning workers
- We just need enough workers to do total length in time available
- $\lceil L/n \rceil$ where $L = \sum \ell_i$
- $O(m)$

Problem G : God's Number (Easy)

- Any solution will do
- Repeating a move three (3) times will undo that move
- Undoing moves in reverse order will solve the cube
- Therefore just reverse the string and triple each character
- $O(|S|)$ where S is the input string

Problem A : Australian vs American (Easy)

- Write a function to check if a date is valid (edge cases, leap days, etc.)
- Since the years are 2000 to 2099, leap years are as simple as the statement implies
- Only six (6) possibilities for date order
- For each order check if all dates are valid in that ordering
- If multiple orders are valid, print UNSURE
- If exactly one is valid, print out which one
- If none are valid, print $\mathrm{IMPOSSIBLE}$
- $O(n)$ (with a constant factor of $3! = 6$)

Problem C : Cryptic Clues (Easyish)

- Two strings are anagrams of each other if they have the same character histogram or sorted characters
- For each candidate word:
  - Build its character histogram (or sort the characters)
  - Try all subranges of words
  - Continue if the number of characters is not the same
  - Continue if the strings are equal
  - Check their histogram (or sorted characters) are the same
- $O(W|S|^2)$ where S is the clue string

Problem D : Dubious Recording (Easy-Medium)

- The sum of all subarrays divided by the number of subarrays
- Number of subarrays is $N(N+1)/2$, make sure N isn't an int or you will overflow
- If there are l elements to the left of a number and r to the right, then it gets counted $(l+1) \times (r+1)$ times
- $O(N)$

Problem K : Krazy Taxi (Easy-Medium)

- We have a directed graph after expanding undirected edges into 2 directed edges
- We want to find out if a node is unreachable, reachable, or only reachable when treating all edges as undirected
- Do a DFS/BFS using the input edges
- Do another DFS/BFS with all directed edges changed to undirected

Problem H : Holiday (Medium)

- So many clarifications about sample 3! :(
- The problem is NOT find the smallest range that contains all temperatures. Read again if you think this. We also feel the first two paragraphs are quite clear about this
- Two monotonic pointers can be used
- Left pointer moves from left to right
- Keep increasing right pointer while you don't have all the temperatures in your set
- Stop when the right pointer reaches the end
- $O(n \log(n))$

Problem J : Juice Machine (Medium)

- The classic coin change problem, except the coins are fractional
- We can reduce to the normal coin change problem and apply DP
- Multiply by the least common multiple of 1 to 10 (2520)
- $O(2520 \times NC)$

Problem I : Island of Love (Medium)

- Seems like union-find (also called onion-find by the judges), except we need to delete edges instead of “union” (or “onion”)
- Let’s just reverse the queries and answer them in reverse
- Now we can just union edges as they are un-deleted
- $O(Q\alpha(F))$

Problem L : Love Actually (Medium-Hard)

- A corner of the triangle must be in a corner of the square
- Try all triangle corners
- Binary search the square side length
- Check by binary searching the rotation angle of the triangle
- Use the law of cosines to get the triangle point locations
- If p is the desired bits of precision, then $O(p^2)$
- Faster solutions exist but I don't like geometry (and I do like binary/ternary search)

Problem B : Bombs Ahoy! (Medium-Hard)

- The mouse can always stay where they are
- BFS but don't go into painted regions
- Make sure to remove previously visited regions that got painted later at the end.
- "Georgie stops moving when the last bomb lands." (Many teams missed this)
- Using a 2D Fenwick tree or segment tree to handle rectangle updates and point look-ups
- $O((NM + K) \times \log(N) \times \log(M))$
- $O((NM + K) \times (\log(N) + \log(M))$ is possible (see next slide)

Problem B : Bombs Ahoy! (Medium-Hard)

- Make a segment tree (or similar) for each row and column
- Mark the perimeter of a paint rectangle by updating 4 segment trees
- Never BFS across painted perimeters
- Remove locations inside any rectangle at the end by doing a sweep of rows from top to bottom and maintaining a segment tree of currently open rectangles