South Pacific ICPC Regionals 2018 Editorial

2018-11-24

Judges

- Andrew Gozzard
- Andrew Haigh
- Bao Truong
- Brandon Fuller
- Connie Pyromallis
- Daniel Anderson
- Darcy Best
- Eliot Courtney
- Evgeni Sergeev
- Henning Koehler
- **Kevin Tran**
- Max Ward-Graham
- Mike Cameron-Jones
- Peter Whalan
- Walter Guttmann
- Xin Wei Chow
- Zac Forman

Problem A : Aligned Typesetting (Easy, 12/12)

- Find sum of world lengths $W = \sum_i w_i$
- Subtract from L to get number of spaces $L' = L - W$
- Distribute them over the $n - 1$ gaps evenly ($L' \equiv 0 \bmod (n-1)$)
- Pitfalls:
  - If $n = 1$, $L' = 0$
  - $L' \geq (n-1)$
- $O(W)$

Problem C : Collecting Stars (Easy, 12/12)

- Sort stars by number of stars required to unlock them
- Add currently reachable stars to priority queue
- Greedily take stars we can currently reach fastest
- As stars are unlocked, add them to the priority queue
- Each star is inserted and removed at most once
- $O(n \log n)$

Problem L: Laps (Easy, 12/12)

- Look at consecutive pairs of locations
- Location decreases: must be in a new lap
- Location same or increases: can be in the same lap
- Count number of times location decreases
- $O(n)$

Problem J: Judging Divisionals (Easy, 12/12)

- Just do it—simulate selection process
- Iterate through the data implementing step 1 and then step 2
- Use maps/dictionaries (or even arrays) to keep track of information
- $O(n)$

Problem D : Dropping Blocks (Medium, 9/12)

- If $a_{i-1} > a_i$, then there *must* be a left interval that ends here.
- Similarly, if $a_{i-1} < a_i$, then there *must* be a right interval that ends here.
- Sweep from left to right and when you hit a right endpoint, remember to subtract off that amount for each pile to your right (when you get there).
- If you ever subtract off those right endpoints and get a negative value, output `NO`. Otherwise, output `YES`.
- $O(N)$

Problem H : Hazardous Driving (Medium, 9/12)

- Find the minimum length path such that the maximum hazard is minimised
- A path from the start to the end using an algorithm like Prim's or Kruskal's to find the min-max hazard (basically, a hazard min-max MST)
- Find the shortest distance path with more hazardous edges removed
- Binary search the hazard is also fine
- $O(M \log N)$

Problem B : Bad Keming (Medium, 7/12)

- We want to find a section of our string that matches every second character of a prefix of our string
- This is just pattern matching on either every odd or every even character against the original string
- We then add all missing characters up to size of original string
  - For $S_0, S_2, \ldots$ maximum match of n gives $2n$
  - For $S_1, S_3, \ldots$ maximum match of n gives $2n + 1$
- We very deliberately broke heuristics and hashing (mod 2^{64})
- Use z-algorithm or KMP
- $O(|S|)$

Problem K : Knights and Dragons (Medium, 6/12)

- Think of each sword as a point in the plane.
- A point is useless if it is some linear combination of other points.
- In other words, a point is useless if it is inside the convex hull (or on the boundary between two other points).
- DO NOT USE FLOATING POINT NUMBERS.
- $O(n \log n)$

- ...did I mention that you should not use floating point numbers...?

Problem F: Far, Far Away (Medium, 2/12)

- Looking for minimal-cost path
- Compute matrix powers using min-plus matrix multiplication
- Binary search to find maximal length with cost below m
- $O(n^3 \log^2 m)$
- $O(n^3 \log m)$ by precomputing matrix powers for powers of 2

Problem I : Interesting World of Arrays (Medium-Hard, 1/12)

- There are n^n arrays to consider (way too many to generate...)
- The sum of the entries in *any* counting array must add up to n (by definition).
- There are $\binom{2n-1}{n-1}$ counting arrays (stars and bars).
- Generate all of counting arrays exhaustively. For each one, you know how many 0s, how many 1s, how many 2s, etc. there were in the original array.
- Do a little combinatorics to compute the number of permutations of those numbers given the desired counting array (modulo m).
- $O\left(\binom{2n}{n} n\right)$

Problem G : Good Cable Management (Hard, 1/12)

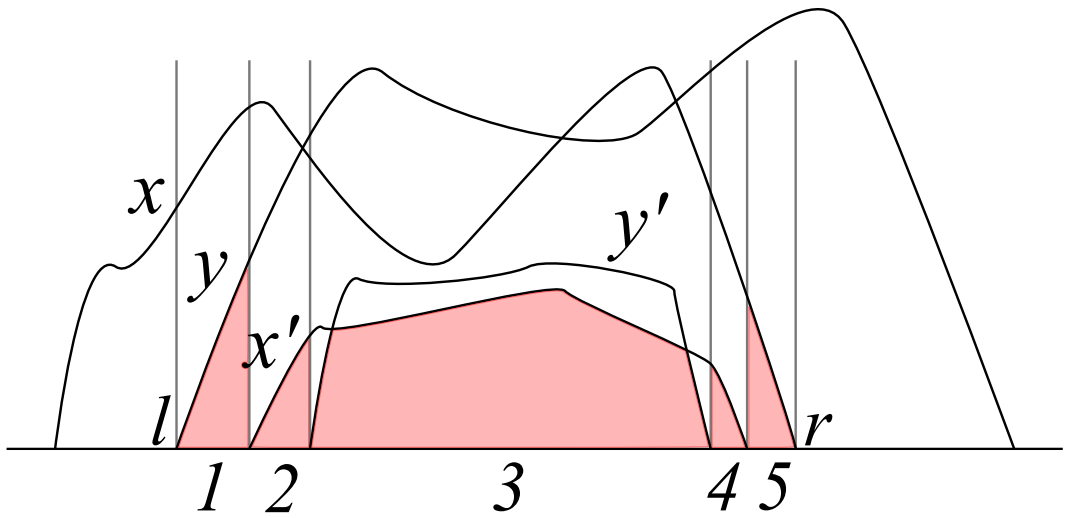
- This problem can be solved by precomputation to efficiently find the lowest common ancestor
- Start with vertex 1 and add an edge and vertex for each new operation
- This forms a tree where every edge is labelled as either "Length" or "Parallel"
- Find the LCA for each query, there are either 2 or 1 outgoing edges. Look at these edges.
- If there is only 1, then it must be length
- If there are 2, then they must both be length, or 1 must be parallel and have label greater than the other (length) edge
- $O((N + Q) \log N)$ (or other LCA complexities)

Problem E : Explosive Wiring (Hard, 0/12)

- This problem is just weighted graph matching

Problem E : Explosive Wiring (Hard, 0/12)

- Just kidding, it isn't that at all
- All the geometry we need: compute all the line segment intersections to determine which wires cross
- Now, some observations:
  - There are some nested regions that are important
  - Call these regions matched regions
  - We can do a DP on matched regions



Problem E : Explosive Wiring (Hard, 0/12)

- A matched region can be defined by at most 2 ceiling wires and left and right endpoints
- The state space is the matched regions, recurrence relation is to try the next pair of crossing wires
- This implies five regions
- This implies an $O(N^6)$ DP which can be improved
- Only the last four regions matter $O(N^5)$
- Don't do both wires in a pair for the recurrence relation at once (with more DP)
- This problem was not currently solved in academia, we are working on a publication now
- $O(N^4)$