

Competition 21 SEP 2019

**Giri Narasimhan**

Programming Team

Fall 2019

# A. Mirror Images

- Trivial

# C. Interpreter

- Involved, but straightforward

# E. Median Updates

- Median =
  - middle item in sorted list (odd length),
  - Or average of middle items (even length)
- Given a dynamic list, i.e., it allows
  - Inserts
  - Deletes
  - Report median after every successful operation

# Augmented RB trees

- Augmented balanced RB tree is an efficient dynamic data structure
- Every insert in this problem involves
  1. Insert into RB tree, and
  2. Computing median in tree (Select)
- Every delete in this problem involves
  1. Delete from RB tree, and
  2. Computing median in tree (Select)

# F. ACM Rank

- Data contains a stream of requests
  - **S** minute teamID problemID result
  - **R** teamID
  - **T** rank
- Goal is to answer each query as efficiently as possible

# Augmented RB Tree

- As with Rank and Select,
  - Augment RB tree with size (of subtree) info
- **S** translates to
  - insert or update operation
- **R** translates to
  - inorder tree traversal
- **T** translates to
  - Doing select operation on augmented tree

# D. Balanced Lineup

- Given sequence of 1,000,000 cow heights (data) & list of 200,000 sublists of cows (queries)
  - Report difference in max and min height in group
  - Every sublist is specified by first and last cow
- Naïve:
  - Search every sublist
  - $10^6 \times 2 \times 10^5 = 2 \times 10^{11}$   $O(Nk)$
- What if we preprocess the data



# Augment RB Trees

- What information to store?
- Can it be maintained?
- How do perform the new operation?

# H. Median on the Plane

- Given  $N$  points on plane
  - $N$  is even
  - No three points are collinear
- Select two points in such a way, that the infinite straight line on which they lie divides all remaining points into two equal-sized sets
- **Naïve**: Try every pair of points and check
  - $O(N^2 \times N) = O(N^3)$

# B. Tree Augmentation

- Examples
  - Tree -> AugTree
- Examples
  - AugTree -> Tree
  - Challenges

# Observation

- Every vertex forms a **star** with its neighbors
- Every **star** in Tree becomes a **clique** in the AugTree
- Thus identify the cliques
- Is that enough
- What about adjacent neighbors in Tree?
  - Cliques with common vertices

# Algorithm for Augmented Tree?

- Use observations above to design an algorithm

# G. Lights