ACM-CP

acmbpdc.org/acm-cp

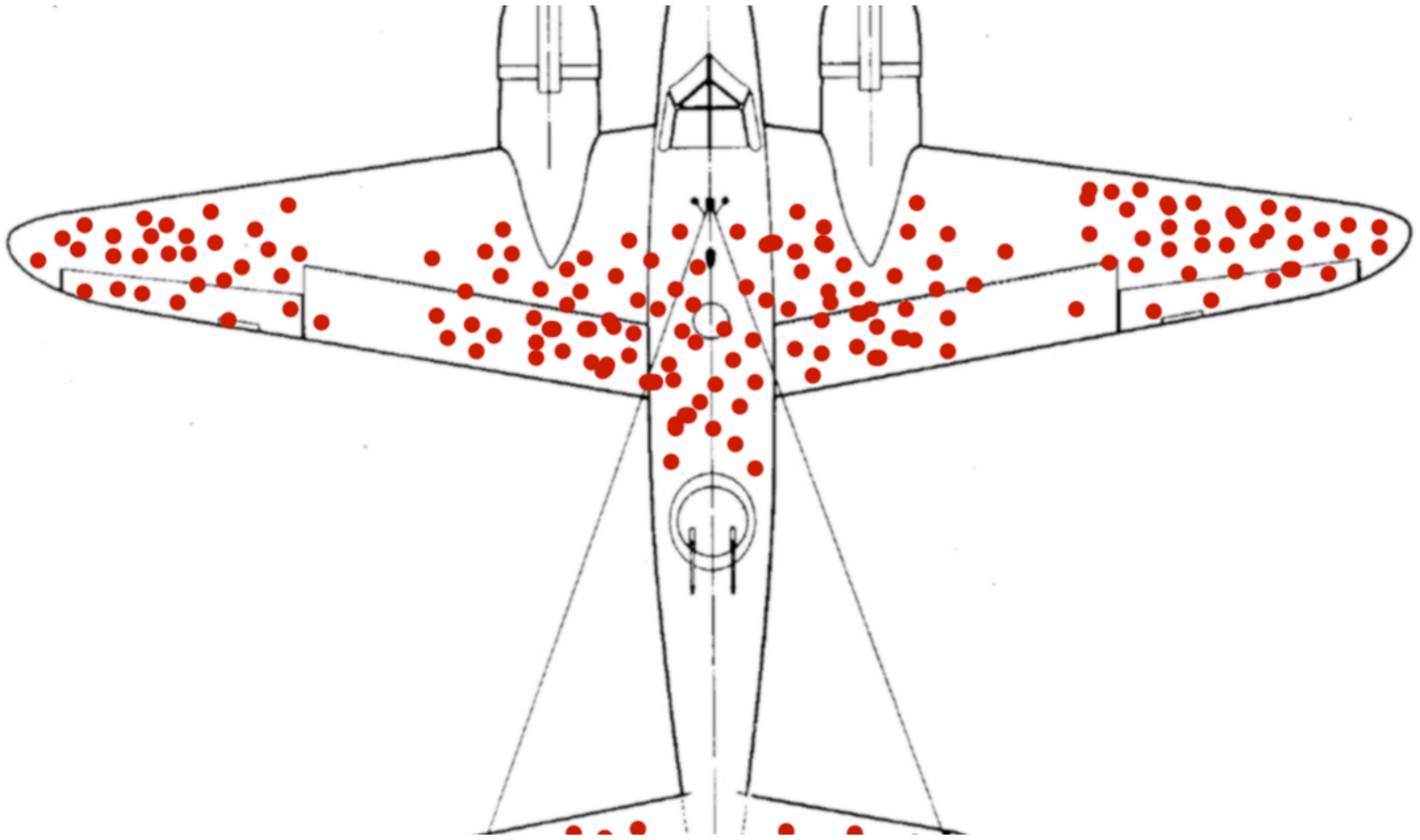
“It takes algorithms to write algorithms”

Is there a way to think?

- During World War II, a mathematician was hired by the Allied forces. His task was to use the samples of planes that returned from the war and provide an analysis as to where more armour had to be put on to these planes to better defend them against the bullets shot by opposing air forces.

- Note that armour cannot be used in all areas of the plane as the allied forces want to strike a balance between the lightness of the aircraft vs. the defence provided by the armour.

- The aggregate of bullets that struck the bodies of all aircrafts in the sample was made and represented in a single figure as shown in the following slide.



Where would you apply the armour to these planes?

Question your assumptions

- The armour needs to be applied at all points where the bullet holes have NOT struck the plane.

- This is because the planes that returned from the war with bullet holes riddled on them are those that survived shots from the enemy.

- Therefore, the areas where there are *no* bullet holes on planes that return from the war signify the points that got hit and damaged by planes that did no return from the war.

- The faulty assumption was that we only considered planes that returned and not the set of all planes that were sent out to battle.

What is competitive programming and how is it different from normal programming?

- Programming ≠ Writing Code
- Programming is 95% problem solving
- Getting to the solution is the toughest part
- Writing code is almost always the easiest part of programming
- Competitive programming is a mind sport
- F1 for programmers
- Fastest person to the finish line (solving all problems) wins

Programming contests

- Set of well-defined problems
  - Input and output formats specified exactly
  - Memory and time constraints
  - Simple command-line programs; no GUI
- Problems need to be solved in stipulated amount of time
- Programs tested on hidden test cases

ACM ICPC

Biggest CP Event

Okay…
So how do I become a reasonably good
competitive programmer?

Tips

- **Type** faster
- **Test** your code before submission
  - Indent your code
  - Test on edge cases, large inputs, etc.
- **Do** algorithm analysis
- **Be** aware of special features and in-built functions present in your programming language

PARTNER UP WITH YOUR BENCH-MATE,
IT’S **CODING** TIME.

Drinks are on the House

Problem Statement

It’s party time at the BPDC-ACM session. And of course, drinks are on the house.

Initially, there are **N** empty glasses. You have **M** instructions of the form **a**, **b**, **k**. For each of these instructions, you pour **k** mL of a magic drink that makes you a ninja programmer into all glasses between **a** to **b** (inclusive). You would like to get the maximum amount of drink. However, you can only pick one glass. Which one will you choose?

Drinks are on the House

Input Format

The first line contains two space-separated integers and **N** and **M**, the number of glasses and the number of instructions.
Each of the next lines contains three space-separated integers **a**, **b** and **k**, the left index, right index and the amount of drink you need to pour.

Constraints

10 ≤ N ≤ 10^7
1 ≤ M ≤ 10^5
1 ≤ a, b ≤ N
1 ≤ k ≤ 10^6

Output Format

Print the glass you should drink from and the amount of drink it has.