Day 4: Problem Analysis

Niyaz Nigmatullin

September 11, 2015

Problem A
Problem B
Problem C
Problem D
Problem E
Problem F
Problem G
Problem H

Problem A. Headshot

Statement

- You are playing russian roulette
- Your opponent already tried to shoot himself, and survived
- What is your strategy either you should rotate the cylinder or not?

Problem A. Headshot

Solution

- Let's compute the probability of surviving, when not rotating
  - Before the first shoot: 01001010
  - Before the second shoot: 01001010
  - Probability to survive: $\frac{C_{00}}{C_0}$
- Let's compute the probability of surviving, when rotating
  - Probability to survive: $\frac{C_0}{n}$

Problem A. Headshot

Solution

- Compare this two numbers as: $\frac{C_{00}}{C_0} < \frac{C_0}{n} \Leftrightarrow C_{00} n < C_0 C_0$
- No need to use `double`
- If you use `double`, check two numbers might be equal, but precision errors make them not equal on the machine

```
int compare(double a, double b) {
    if (Math.abs(a - b) < 1e-8)
        reutrn 0;
    return a < b ? -1 : 1;
}
```

Problem B. Alien communication masterclass

Statement

- Compose an expression which:
  - Holds in the given bases
  - Doesn't hold in all other bases

Solution

- Next expression is zero only in k base

$$10\underbrace{-1-1-\cdots-1}_{\text{k subtractions}}=0$$

Problem B. Alien communication masterclass

$$(10\underbrace{-1-1-\cdots-1}_{a_1 \text{ subtractions}})\times$$

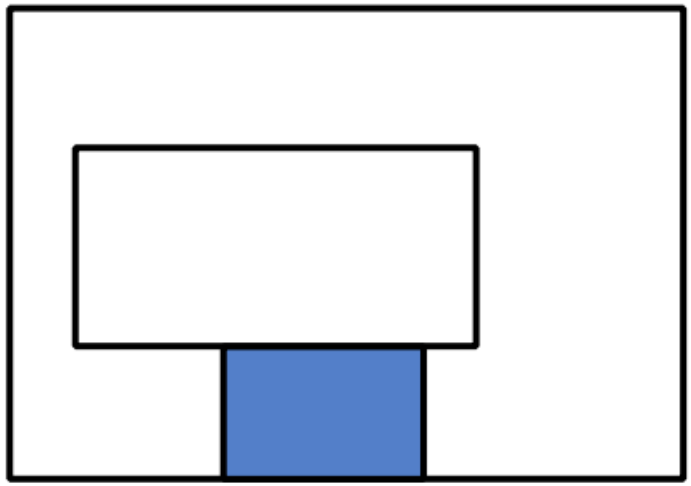
$$\times(10\underbrace{-1-1-\cdots-1}_{a_2 \text{ subtractions}})\times\cdots\times$$

$$\times(10\underbrace{-1-1-\cdots-1}_{a_N \text{ subtractions}})=0$$

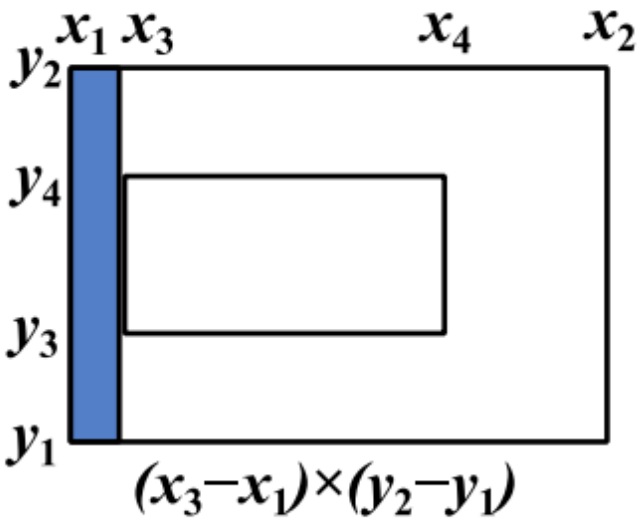
Problem C. Grave

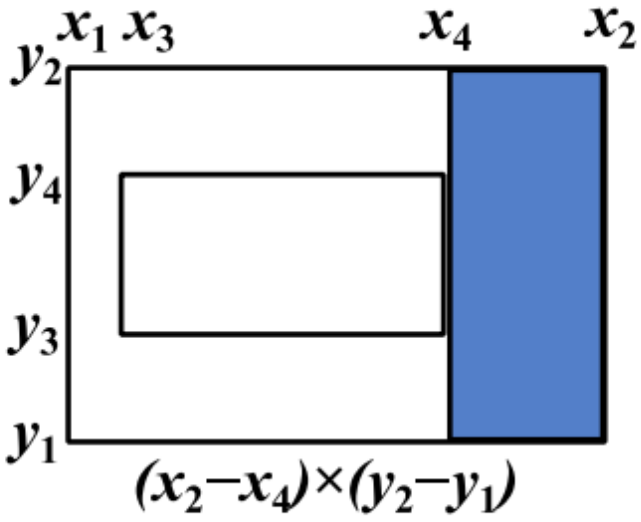
Statement

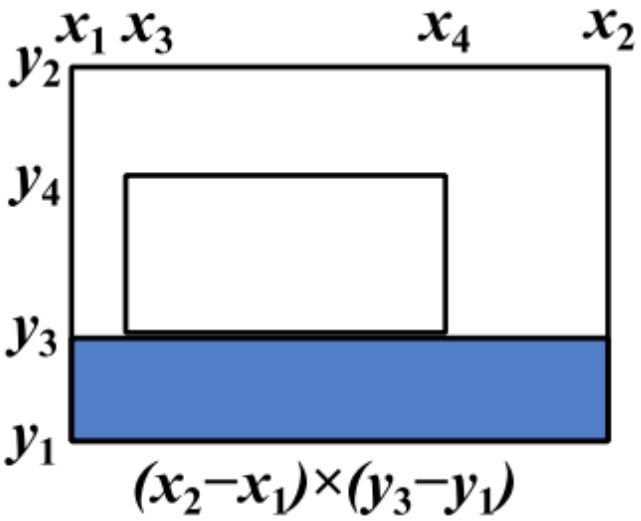
- You are given a rectangle with a rectangular hole
- You are to place $w \times h$ rectangle inside it

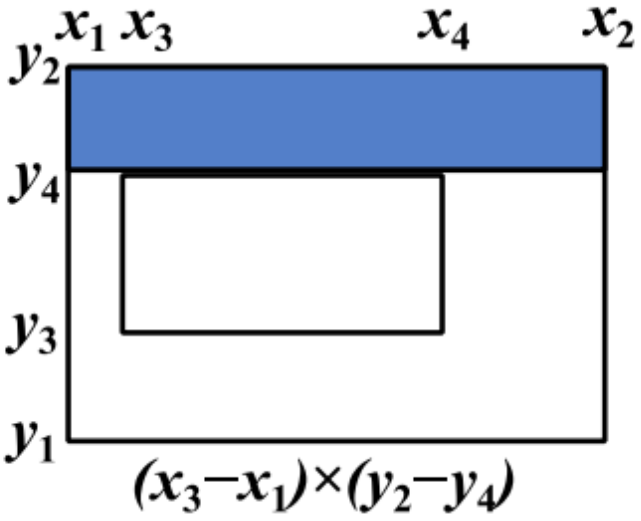


Problem C. Grave









Problem C. Grave

Solution

- New rectangle has to fit into one of maximal rectangles
- Time complexity: $O(1)$

Problem D. Bureacracy

Statement

- Find not cancelled laws
- May cancel a canceling law

Solution

- Starting from the last statement
- Active
  - Apply an action
- Canceled
  - Skip

Problem D. Bureacracy

Run example

- 1. declare
- 2. cancel 1
- 3. declare
- 4. cancel 2
- 5. cancel 3

Problem D. Bureacracy

Run example

- 1. declare
- 2. cancel 1
- 3. declare
- 4. cancel 2
- 5. cancel 3

Problem D. Bureacracy

Run example

- 1. declare
- 2. cancel 1
- 3. declare
- 4. cancel 2
- 5. cancel 3

Problem D. Bureacracy

Run example

- 1. declare
- 2. cancel 1
- 3. declare
- 4. cancel 2
- 5. cancel 3

Problem D. Bureacracy

Run example

- 1. declare
- 2. cancel 1
- 3. declare
- 4. cancel 2
- 5. cancel 3

Problem E. Ideal contest

Statement

- Count the contest characteristics

The contest header may contain arbitrary number of lines

Team	A	B	C	D	E	=	Time	R
Revda STU	+	+	+2	+1	-9	4	9274	1
Girvas NU #1	+	+	-1	.	-11	2	321	2
Kargopol SU	+	-3	+	.	-4	2	321	2
Utorgosh SU	.	.	.	+	-5	1	122	4
Dubrovno SU	.	+	-1	.	-4	1	123	5
Girvas NU - 2	.	.	.	-5	-99	0	0	6

Problem E. Ideal contest

Weird things

The contest header may contain arbitrary number of lines

Team	A	B	C	D	E	=	Time	R
Revda STU	+	+	+2	+1	-9	4	9274	1
Girvas NU #1	+	(+)	-1	.	-11	2	321	2
Kargopol SU	+	-3	(+)	.	-4	2	321	2
Utorgosh SU	.	.	.	+	-5	1	122	4
Dubrovno SU	.	+	-1	.	(-4)	1	123	5
Girvas NU (−) 2	.	.	.	-5	-99	0	0	6

Problem E. Ideal contest

Header

- Search for header
  - Problem letters
  - First word is Team
  - Line of minuses
- Number of problems
  - From the header

Table body

- Tokenize by spaces
- Look from the end
  - Use the number of problems

Problem E. Ideal contest

Vainness

- Teams who didn't solve anything

Oversimplification

- Teams who solved everything

Evenness

- Spaces in the number of solved problems

Unsolvability

- Problems not solved by anyone

Problem E. Ideal contest

Instability

- Teams who took a higher place, but didn't solve a problem

Team	C	=	Time	R
Revda STU	+2	4	9274	1
Girvas NU #1	-1	2	321	2
Kargopol SU	+	2	321	2

Problem E. Ideal contest

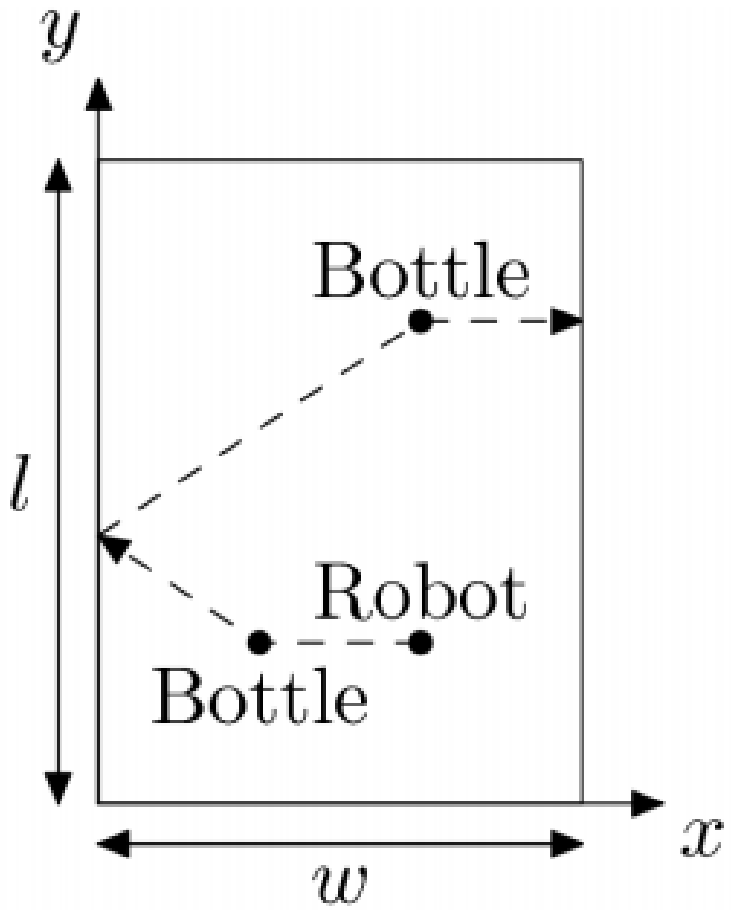
Negidealness

- Formula: $1.030\ \mathrm{V} + 3.141\ \mathrm{O} +$
- $2.171\ \mathrm{E} + 1.414\ \mathrm{U} +$
- $(I_1 + I_2 + \cdots + I_P)/P$

Problem F. Kitchen Robot

Statement

- Remove all the bottles
  - As soon as possible



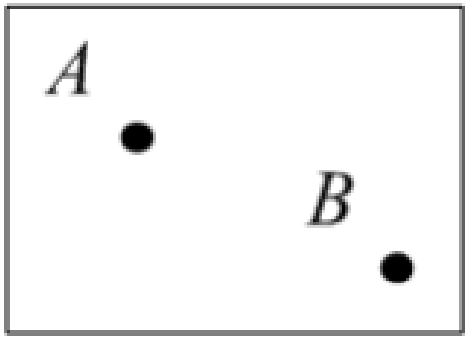
Problem F. Kitchen Robot

Start and finish

- Start
  - To the first bottle
  - Straightforward
- Finish
  - From the last bottle
  - To the nearest boundary

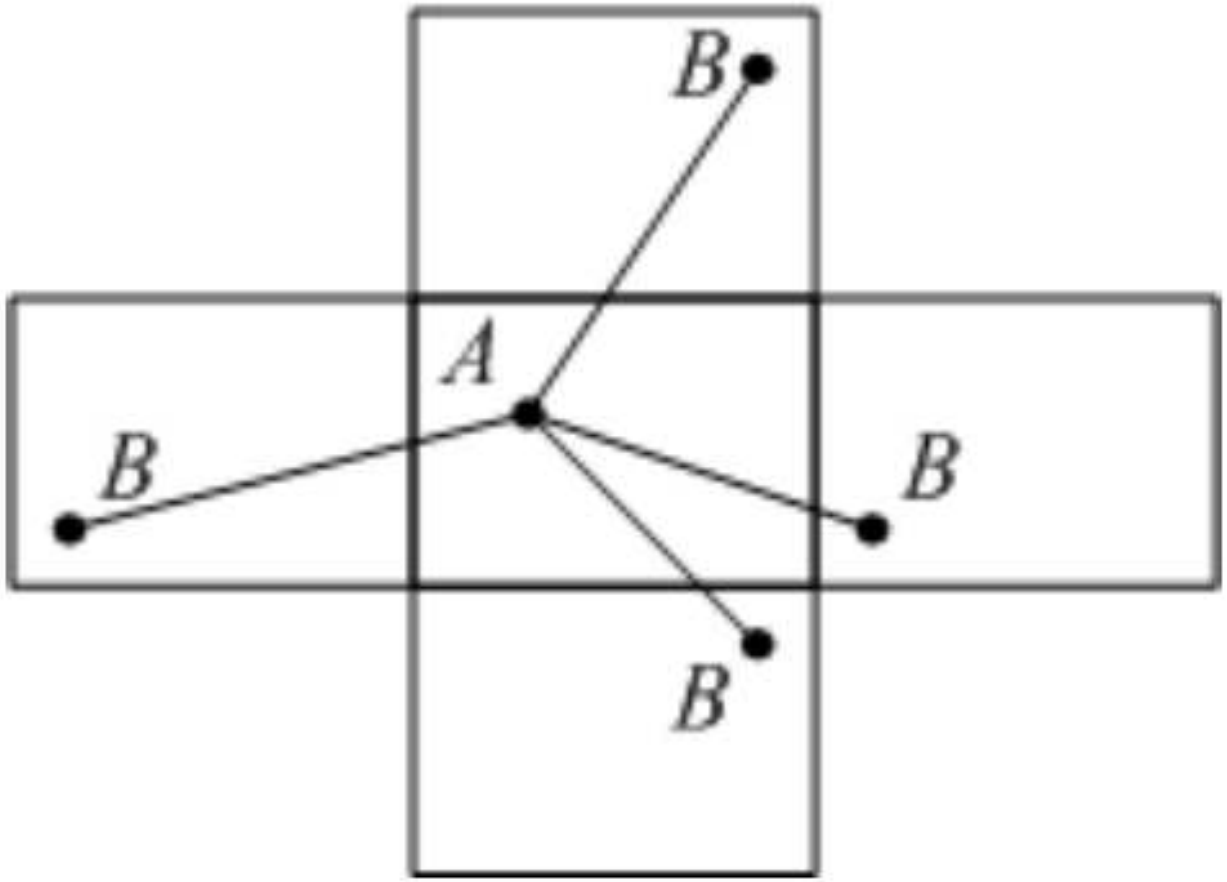
Problem F. Kitchen Robot

Between the bottles



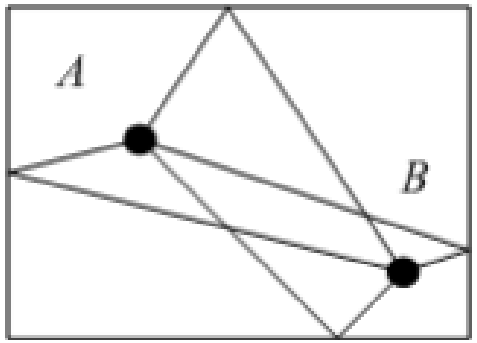
Problem F. Kitchen Robot

Between the bottles



Problem F. Kitchen Robot

Fold up



Problem F. Kitchen Robot

Solution

- Transition lengths are known
  - Traveling salesman problem
- Dynamic programming on subsets
  - `F[A][v]` — shortest path to get to bottle v, collecting all the bottles from set A
  - To compute: try all the bottles u as the previous one, and exclude bottle v from the set
  - Time complexity: $O(2^n \cdot n^2)$

Problem G. Asteroids

Statement

- You are given two polyhedra
- Find the minimum distance between their centers of mass

Solution

- Find the faces of each polyhedron
- Iterate over all triples of points, if all other points are on one side of the plane, then it's a face

Problem G. Asteroids

Finding center of mass

- Choose one vertex
- Choose all triples of consecutive points from all the faces
- Polyhedron broke into tetrahedrons
- Center of mass of tetrahedron is just the coordinate average of points
- Assume that all mass of each tetrahedron is located in its center of mass
- Now find the mass average of these points, assuming that mass is proportional to the volume

Problem G. Asteroids

Finding minimum distance

- Find minimum distance from center of mass to outside of polyhedron
- Just minimum of all distances to all the faces as planes

Problem H. Galaxy interconnection

Statement

- Given a graph
  - Contains a cycle of size k
- Paint in k colors
  - Neighbors have different colors
- From each vertex there should be a path of k vertices of k different colors

Problem H. Galaxy interconnection

Solution

- Consider a certain order of colors
- Paint the cycle in k colors in order
- DFS the other vertices from the cycle
  - Paint in some unused color which goes (in order) after a used one
- For unvisited vertices choose the color in order