











2020 ICPC Vietnam National

Final Scoreboard

RK	TEAM		SLV.	TIME	A	B	C	D	E	F	G	H	I	J	K	L	M
1	EggCentroy		13	1867	1 19	2 114	1 96	2 202	1 185	3 268	2 240	1 137	2 36	1 90	1 125	1 160	2 55
2	HCMUS-Illuminate		11	1284	1 13	3 141	1 163	2 266			1 232	1 113	1 63	1 27	1 79	2 91	1 16
3	alpha		11	1939	1 11	1 193	5 167		2 239	5 ---	3 268	1 82	3 41	1 114	1 233	10 177	1 54
4	HCMUS-CentralReborn		10	1454	2 13	3 221	1 192	3 162	5 285			1 72	2 59	2 84	1 103	8 --	2 23
5	HCMUS-CalicoCat		9	1038	1 12	3 92	3 217	4 250			1 --	1 50	1 24	2 124	1 71	1 --	1 38
6	C'est BON!		9	1204	1 13	3 --	3 157	1 257	1 251		2 --	2 76	2 86	2 99	1 74	3 --	2 71
7	A*		9	1316	2 11	1 137	6 --	2 198	4 282			3 208	1 90	1 111	1 118		1 21
8	Aho Corasick		9	1534	1 12	3 255	6 293				3 223	1 50	2 133	1 187	3 122		1 19
9	Semiperfect		9	1608	1 23	3 131	1 35	7 298				4 282	1 81	1 170	1 164		6 104
10	gamma		8	852	2 11	1 35	1 --				3 256	4 116	2 128	1 72	1 55		1 39

Problems

	NAME	SOLVED	FIRST AC	DIFFICULTY
A	Astrological Sign	320	PTIT.#1f1e33 (5)	0
B	Battle of Hogwarts	53	gamma (35)	2.5
C	Cable Car	21	Semiperfect (35)	4
D	Dividing Kingdom	16	HCMUS-CentralReborn (162)	2.5
E	Easy Arithmetic	8	CSP PBC02 (147)	2.5
F	Fluffy Cat	1	EggCentroy (268)	3.5
G	Greatest Pair	5	Aho Corasick (223)	3
H	Highway to Mount Fansipan	138	Karatsubaaaaa (41)	1.5
I	Infinite 2D Array	92	HCMUS-CalicoCat (24)	2.5
J	Just Enough Water	108	HCMUS-Illuminate (27)	1.5
K	Kingdom of Cats	140	PTIT.#1f1e33 (31)	1
L	Looping Around	4	HCMUS-Illuminate (91)	2.5
M	Malfunctioning Robot	293	Bogosort (11)	0.5

A - Astrological Sign

- Độ khó: 0*
- Số AC:
- Đề bài:
 - Cho ngày & tháng. Tìm cung hoàng đạo

Symbol	Astrological Sign names	Birthday
♈	Aries	March 21 – April 20
♉	Taurus	April 21 – May 20
♊	Gemini	May 21 – June 21
♋	Cancer	June 22 – July 22
♌	Leo	July 23 – August 22
♍	Virgo	August 23 – September 21
♎	Libra	September 22 – October 22
♏	Scorpio	October 23 – November 22
♐	Sagittarius	November 23 – December 21
♑	Capricorn	December 22 – January 20
♒	Aquarius	January 21 – February 19
♓	Pisces	February 20 – March 20

A - Astrological Sign

```
if ((month == "Mar" && day >= 21) || (month == "Apr" && day <= 20)) cout << "Aries" << endl;
if ((month == "Apr" && day >= 21) || (month == "May" && day <= 20)) cout << "Taurus" << endl;
if ((month == "May" && day >= 21) || (month == "Jun" && day <= 21)) cout << "Gemini" << endl;
if ((month == "Jun" && day >= 22) || (month == "Jul" && day <= 22)) cout << "Cancer" << endl;
if ((month == "Jul" && day >= 23) || (month == "Aug" && day <= 22)) cout << "Leo" << endl;
if ((month == "Aug" && day >= 23) || (month == "Sep" && day <= 21)) cout << "Virgo" << endl;
if ((month == "Sep" && day >= 22) || (month == "Oct" && day <= 22)) cout << "Libra" << endl;
if ((month == "Oct" && day >= 23) || (month == "Nov" && day <= 22)) cout << "Scorpio" << endl;
if ((month == "Nov" && day >= 23) || (month == "Dec" && day <= 21)) cout << "Sagittarius" << endl;
if ((month == "Dec" && day >= 22) || (month == "Jan" && day <= 20)) cout << "Capricorn" << endl;
if ((month == "Jan" && day >= 21) || (month == "Feb" && day <= 19)) cout << "Aquarius" << endl;
if ((month == "Feb" && day >= 20) || (month == "Mar" && day <= 20)) cout << "Pisces" << endl;
```

B - Battle of Hogwarts

- Độ khó: 2.5*
- Số AC:
- Đề bài:
 - Cho bảng với 3 loại ô (cấm / trống / trống đặc biệt)
 - Tìm cách chặn ít ô trống nhất để không có đường từ (1, 1) đến (R, C)

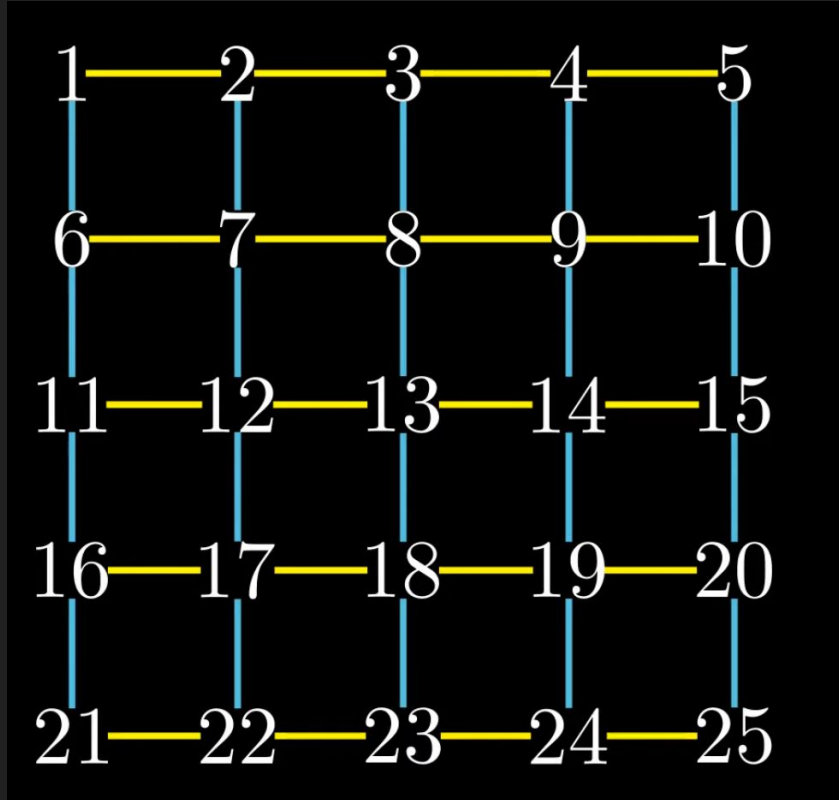
B - Battle of Hogwarts

- → tìm đường đi từ cạnh trái dưới đến cạnh phải trên
 - Nếu qua các ô "magic immune" → chi phí bằng 0
 - Qua các ô "normal" → chi phí bằng 1
- Thuật toán:
 - Dijkstra: $O(R*C*\log)$
 - BFS 0-1: $O(R*C)$

C - Cable Car

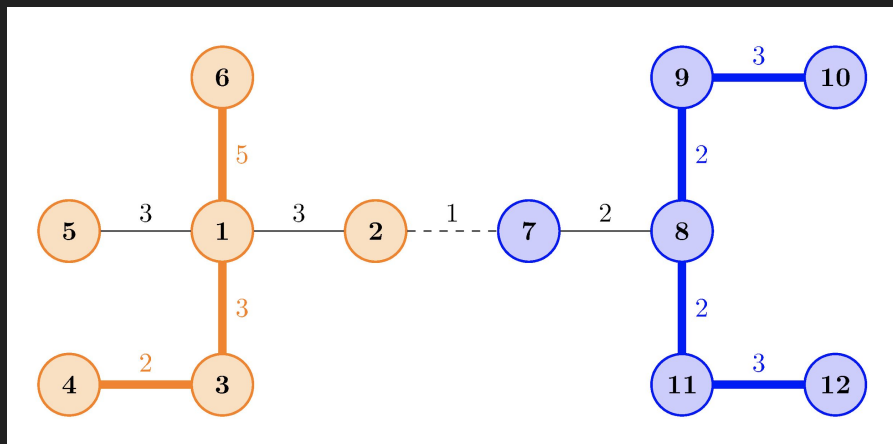
- Độ khó: 4*
- Số AC:
- Đề bài:
 - Xây dựng đồ thị
 - 2 loại cạnh
 - Mỗi loại gồm 1 vài bamboo
 - Không có 1 cặp u v liên thông với cả 2 loại cạnh

C - Cable Car



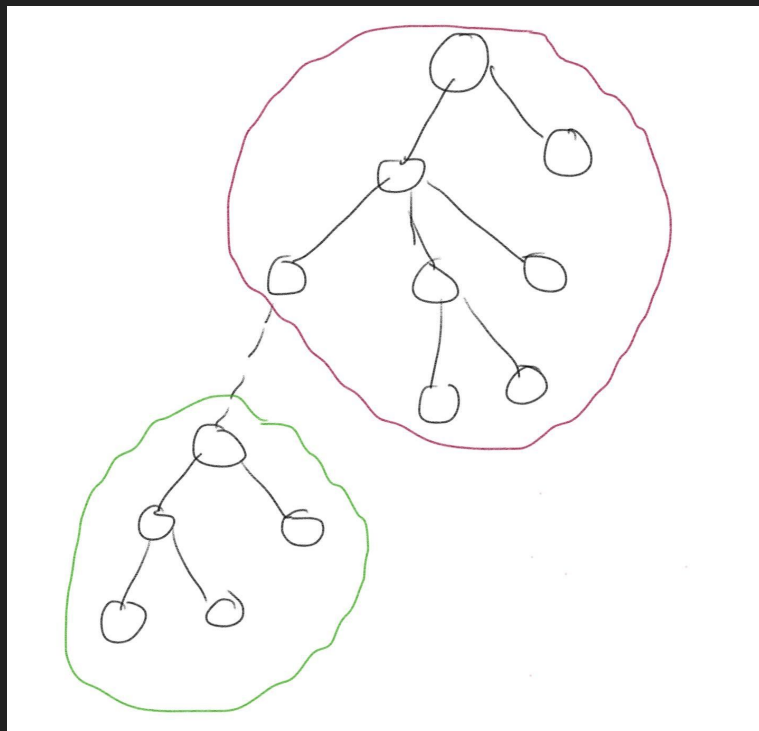
D - Dividing Kingdom

- Độ khó: 2.5*
- Số AC:
- Đề bài:
 - Cắt cây làm 2 cây
 - Chênh lệch đường kính nhỏ nhất



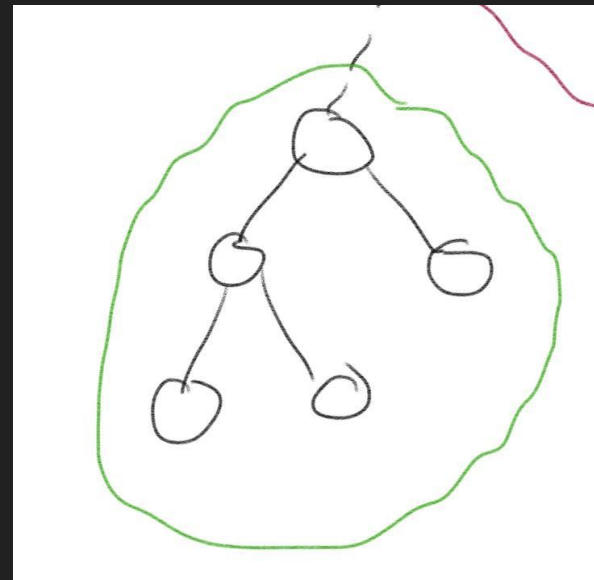
D - Dividing Kingdom

- QHĐ trên cây
- Cắt cạnh v-u:
 - $T(u)$ = cây con gốc u
 - $R(u)$ = (cây ban đầu) - $T(u)$
- → cần tính đường kính của:
 - $T(u)$
 - $R(u)$
 - với mọi u



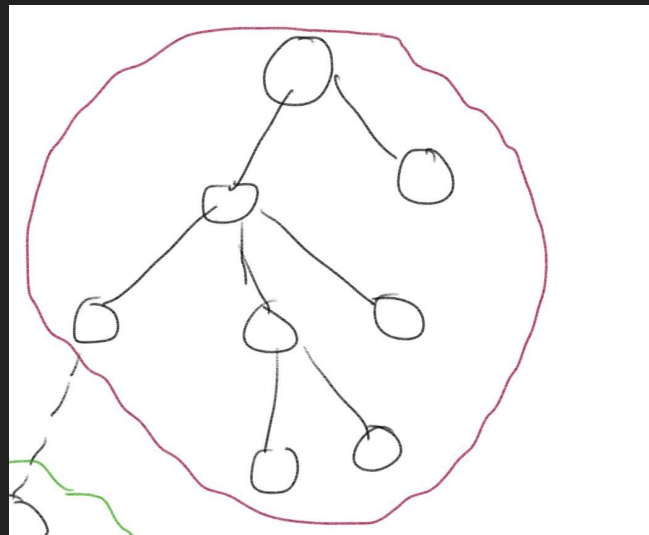
D - Dividing Kingdom

- $T(u)$
 - $\text{diam}(u)$ = đường kính cây con gốc u
 - $\text{down}(u, k)$ = đường đi dài nhất
 - trong $T(u)$,
 - xuất phát ở u
 - $k \leq 2$
- Con u : v_1, v_2, \dots
 - \rightarrow Tính $\text{down}(u, k)$ theo $\text{down}(v_i, k)$
 - \rightarrow Tính $\text{diam}(u)$ theo $\text{diam}(v_i)$, $\text{down}(u)$



D - Dividing Kingdom

- $R(u)$
 - $up_diam(u)$ = đường kính $R(u)$
 - $down(u)$ = đường đi dài nhất:
 - Trong $T(u)$
 - Xuất phát từ u
 - $up(u)$ = đường đi dài nhất:
 - Trong $R(u)$
 - Xuất phát từ u
- Cha $u = fu$
 - \rightarrow Tính $up(u)$ theo $up(fu)$ và $down(fu, k)$
 - \rightarrow Tính $up_diam(u)$ theo
 - $up_diam(fu)$
 - $up(u), down(fu)$

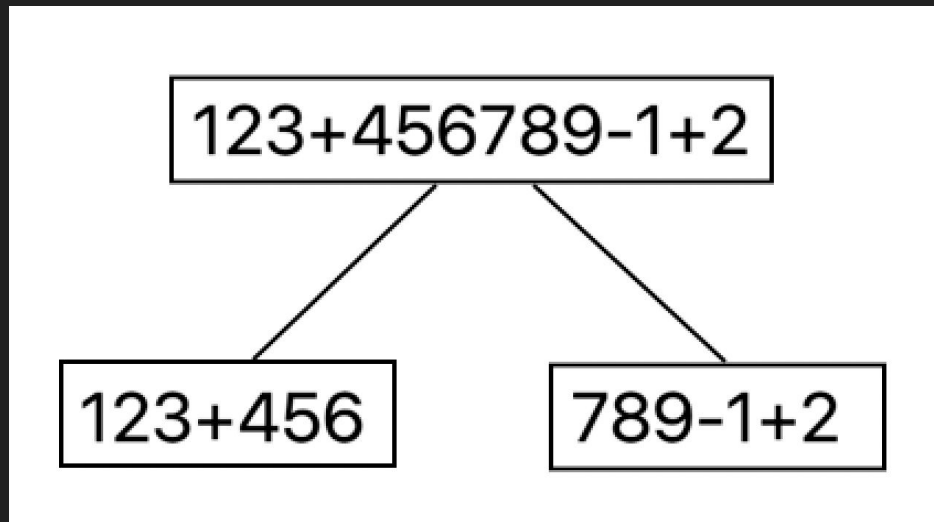


E - Easy Arithmetic

- Độ khó: 2.5*
- Số AC:
- Đề bài:
 - Cho biểu thức dài miên man
 - Truy vấn:
 - Thay đổi 1 ký tự
 - Tính biểu thức từ L đến R

E - Easy Arithmetic

- Segment Tree
- Mỗi nút quản lý 1 đoạn biểu thức



E - Easy Arithmetic

```
63 struct Node {
64     int value, firstValue, lastValue;
65     bool allDigit;
66     int firstLength, lastSign;
67
68     Node() {
69         value = firstValue = lastValue = 0;
70         allDigit = true;
71         firstLength = 0;
72         lastSign = 1;
73     }
```

F - Fluffy Cat

- Độ khó: 3.5*
- Số AC:
- Đề bài:
 - Interactive
 - Mèo bắt chuột
 - Mèo đi 2 bước, chuột đi 1
 - Không biết vị trí, chỉ biết khoảng cách

G - Greatest Pair

- Độ khó: 3*
- Số AC:
- Đề bài:
 - Cho cây N đỉnh
 - Tìm cặp (u, v) có $\max \gcd(\text{label}(u), \text{label}(v)) * \text{dist}(u, v)$

G - Greatest Pair

- Với mỗi ước chung g , xét các đỉnh chia hết cho g
 - Số nhiều ước nhất có khoảng 300 ước
 - → Mỗi đỉnh xét không quá 300 lần
 - → Tổng: Không quá $300 \cdot N$
- → cho 1 tập đỉnh, tìm đường kính
 - $O(N)$

H - Highway to Mount Fansipan

- Độ khó: 1.5*
- Số AC:
- Đề bài:
 - Đếm số ô chữ

I	N	V	I	T	E
C	A	T			
P	A	R	T	Y	
C	O	W			

H - Highway to Mount Fansipan

- Độ dài từ ≤ 50
 - Chỉ cần quan tâm $N \leq 50$
- Duyệt cột dọc:
 - $O(D)$ (D = dictionary size)
 - Mỗi cột dọc: xử lý $O(N)$
- → Biết cần bao nhiêu từ thỏa mãn:
 - Chữ cái đầu tiên = c
 - Độ dài = l
 - → $\text{product}(C(\text{in_dictionary}, \text{need}))$

I	N	V	I	T	E
C	A	T			
P	A	R	T	Y	
C	O	W			

I - Infinite 2D Array

- Độ khó: 2.5*
- Số AC:
- Đề bài:
 - Tính $F(x, y)$

- $F_{0,0} = 0,$
- $F_{0,1} = F_{1,0} = 1,$
- For $i \geq 2, F_{i,0} = F_{i-1,0} + F_{i-2,0},$
- For $i \geq 2, F_{0,i} = F_{0,i-1} + F_{0,i-2},$
- For $i, j \geq 1, F_{i,j} = F_{i-1,j} + F_{i,j-1}.$

Here are the first few values of F :

	0	1	2	3	4	5	6
0	0	1	1	2	3	5	8
1	1	2	3	5	8	13	21
2	1	3	6	11	19	32	53
3	2	5	11	22	41	73	126
4	3	8	19	41	82	155	281
5	5	13	32	73	155	310	591
6	8	21	53	126	281	591	1182

Invasion Plan

Problem ID: invasionplan

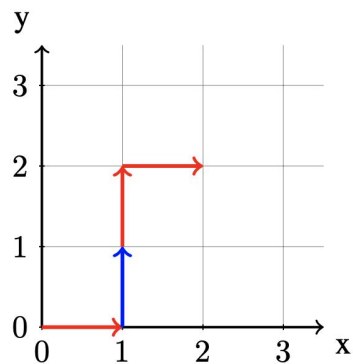
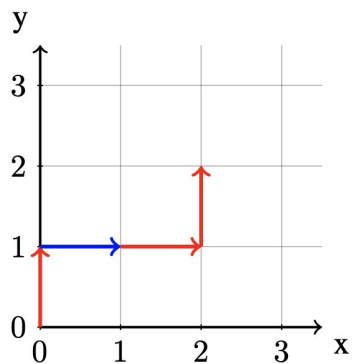
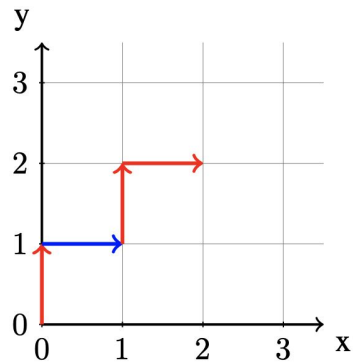
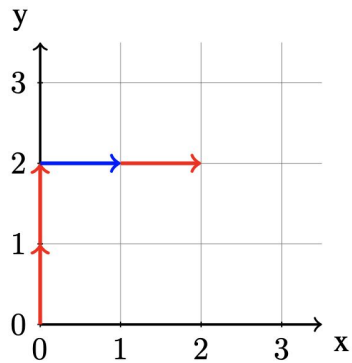
The army is launching an attack on DK - a well known gangster who is responsible for a lot of dirty business. DK lives in a remote location. His land is square shaped, bounded by 4 segments, $y = 0$, $x = 0$, $y = 10^6$ and $x = 10^6$. There are 4 high walls along the boundary.

The target is a mansion located at (x_0, y_0) ($0 < x_0, y_0 < 10^6$). The North and East side of this area is inaccessible so the troops have to start at $(0, 0)$, right outside of the area. The invasion will consist of 3 phrases:

- In the first phrase, they will choose a direction North (going along the West wall) or East (going along the South wall). Each move, they will move 1 or 2 units ahead.
- At location $(x_p, 0)$ or $(0, y_p)$ (depends on the direction in the first phrase), they will climb the wall to break into the land. If they climb over the South wall at $(x_p, 0)$, they will end up at $(x_p, 1)$; if they climb over the West wall at $(0, y_p)$, they will end up at $(1, y_p)$.
- After entering the gangster land, the troops will move toward the mansion only using North-bound or East-bound moves. In this phrase, each step they can only move 1 unit ahead and they can change direction after each step.

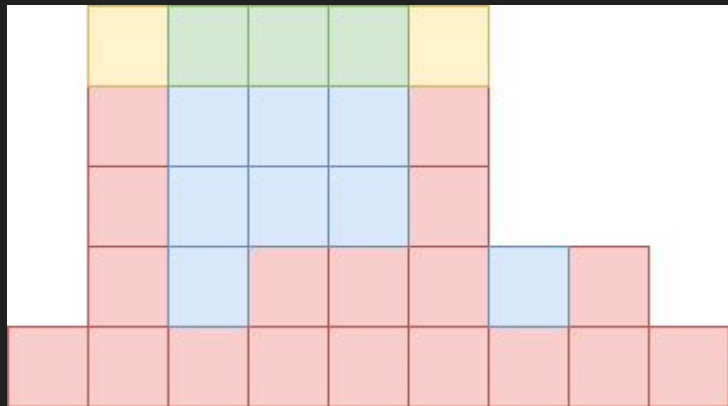
Your task is to count the number of possible movements from $(0, 0)$ to (x_0, y_0) .

I - Infinite 2D Array



J - Just Enough Water

- Độ khó: 1.5*
- Số AC:
- Đề bài:
 - Cho độ cao N cột
 - Có thể tăng 1 số cột lên 1 đơn vị (không quá K lần)
 - Tìm lượng nước đọng lại lớn nhất

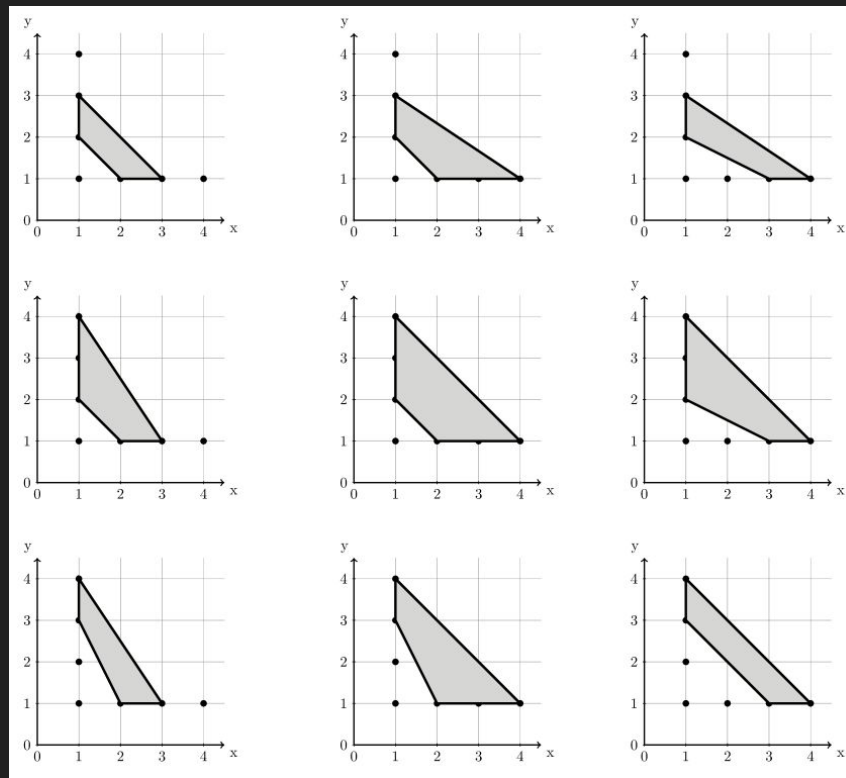


J - Just Enough Water

- Chỉ cần tăng độ cao của tối đa 3 cột
 - Nếu tăng 4 cột → có thể chỉ tăng 2/3 cột
- For 3 cột cần tăng: $O(N^3)$
 - For số lần tăng 2 cột: $O(K^2)$
 - Tính độ cao mới N cột: $O(1)$
 - Tính lượng nước: $O(N)$
- → $O(N^4 * K^2)$

K - Kingdom of Cats

- Độ khó: 1*
- Số AC:
- Đề bài:
 - Tìm bộ 4 điểm tạo thành tứ giác lồi



K - Kingdom of Cats

- For 4 điểm: $O(N^4)$
- Kiểm tra tứ giác lồi: $O(1)$

```
65     auto inside = [&] (pi a, pi b, pi c, pi d) {
66         return area(a, b, c) == area(a, b, d) + area(b, c, d) + area(c, a, d);
67     };
68     int res = 0;
69     FOR(i, 0, n) FOR(j, i + 1, n) FOR(k, j + 1, n) FOR(l, k + 1, n) {
70         auto [a, b, c, d] = make_tuple(pts[i], pts[j], pts[k], pts[l]);
71         if (!inside(a, b, c, d) && !inside(b, c, d, a) && !inside(c, d, a, b) && !inside(d, a, b, c)) {
72             res++;
73         }
74     }
75     cout << res << "\n";
```

L - Looping Around

- Độ khó: 2.5*
- Số AC:
- Đề bài:
 - Tìm đường gấp khúc kín không tự cắt đi qua N điểm cho trước

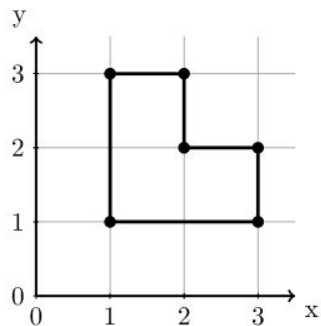


Fig. 1

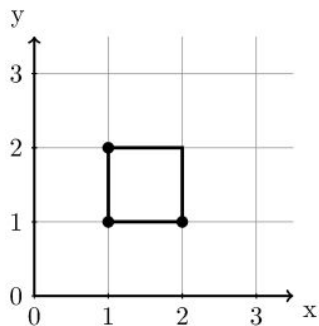


Fig. 2

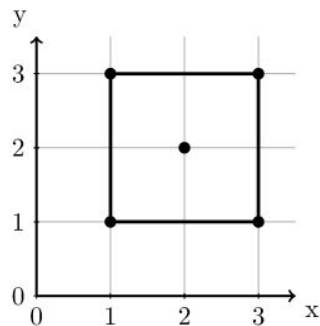


Fig. 3

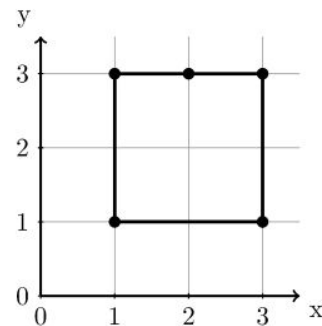
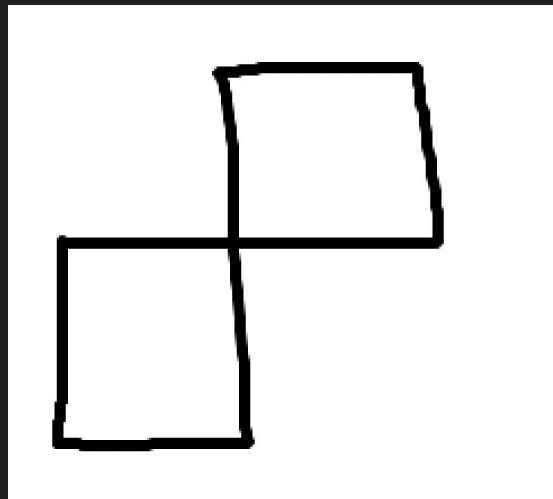


Fig. 4

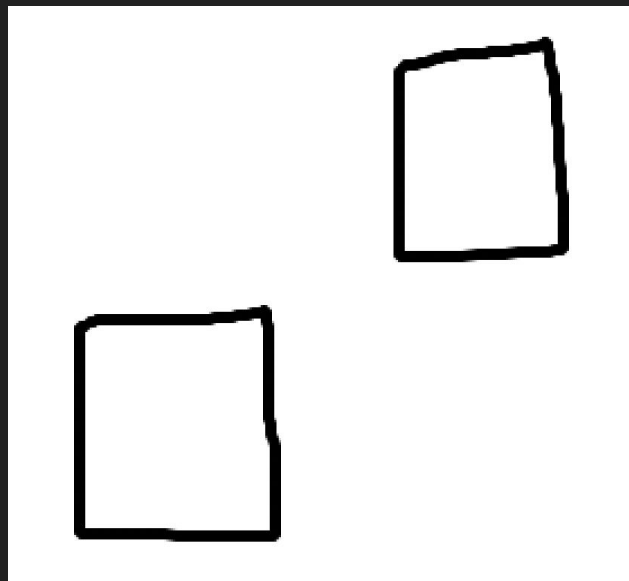
L - Looping Around

- Với mỗi tọa độ X, có chẵn điểm
- Với mỗi tọa độ Y, có chẵn điểm



L - Looping Around

- Với mỗi tọa độ X, có chẵn điểm
- Với mỗi tọa độ Y, có chẵn điểm
- Không có 2 cạnh nào cắt nhau



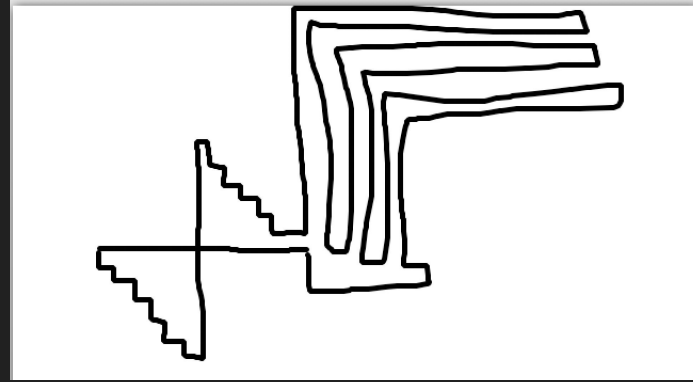
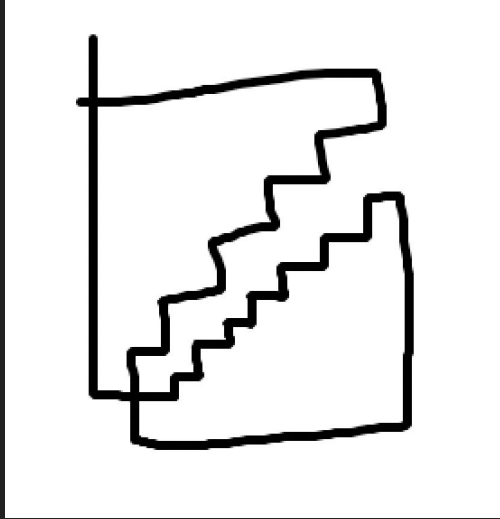
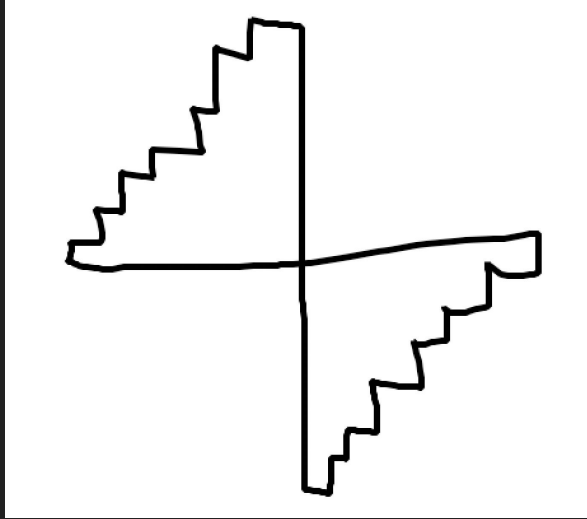
L - Looping Around

- Với mỗi tọa độ X, có chẵn điểm
- Với mỗi tọa độ Y, có chẵn điểm
- Không có 2 cạnh nào cắt nhau
- 1 thành phần liên thông
- \Rightarrow AC

L - Looping Around

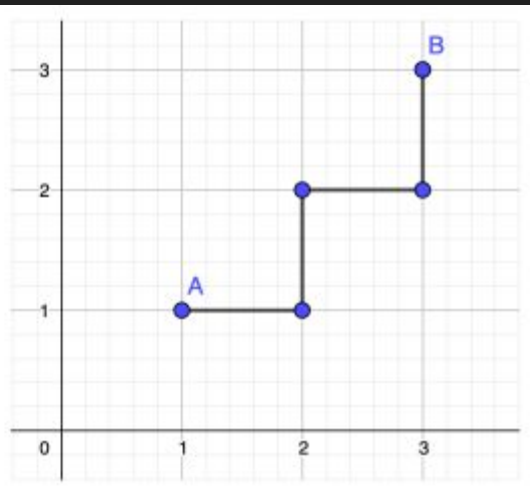
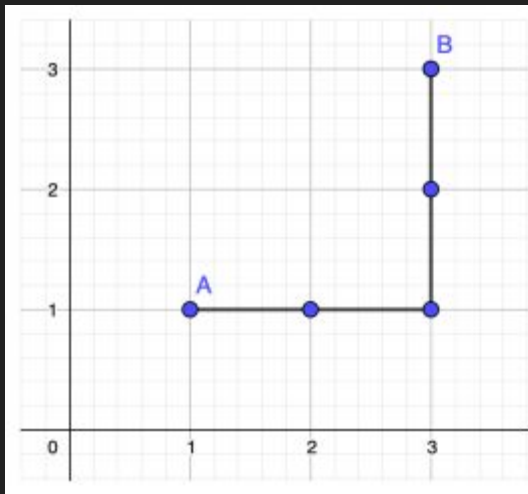
- Với mỗi tọa độ X, có chẵn điểm
 - $O(N \cdot \log)$
- Với mỗi tọa độ Y, có chẵn điểm
 - $O(N \cdot \log)$
- Không có 2 cạnh nào cắt nhau
 - $O(N \cdot \log)$
- 1 thành phần liên thông
 - $O(N)$
- $\Rightarrow AC$

L - Looping Around



M - Malfunctioning Robot

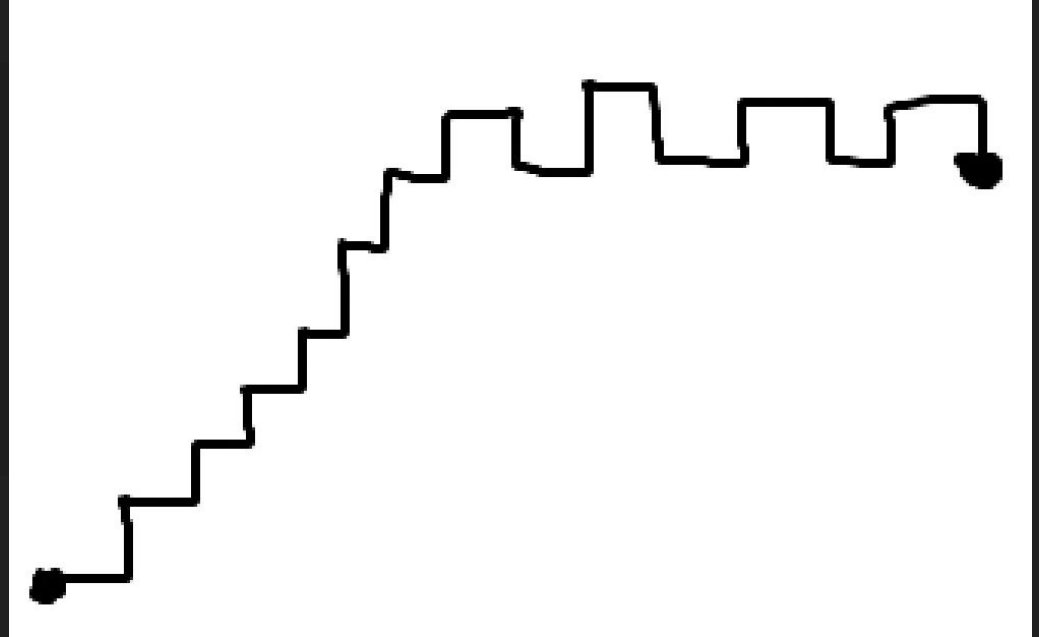
- Độ khó: 0.5*
- Số AC:
- Đề bài:
 - Tìm đường đi ngắn nhất giữa (x_1, y_1) và (x_2, y_2) , không được đi cùng hướng 2 lần liên tiếp



M - Malfunctioning Robot

```
cin >> x1 >> y1 >> x2 >> y2;

long long dx = llabs(x1 - x2);
long long dy = llabs(y1 - y2);
if (dx % 2 == dy % 2) {
    cout << max(dx, dy) * 2 << endl;
} else {
    cout << max(dx, dy) * 2 - 1 << endl;
}
```



Q & A

General, 14:01:23, from Team 227: alpha

Cam on vi 1 ky thi bo ich va ly thu! <3

[Reply](#)