

# CSES swap game

## 題意

- 給你一個  $3 \times 3$  grid 如下

```
1 | 1 2 3
2 | 4 5 6
3 | 7 8 9
```

- 問你給你另一個也是由  $1 \sim 9$  組成的 grid 從原來的變成現在的要最少多少操作
- 每次操作可以 **swap** 相鄰兩個數

## 想法

- $3 \times 3 \Rightarrow$  最多也才 9 皆乘種狀態
  - 暴力 BFS
- 需 grid 壓成一個數字
  - hash
  - $G = \sum_{i=0}^8 a_i \times 9^i$
  - 其中  $a_i$  為把位置在  $(r, c) = (\frac{i}{3}, i\%3)$  (其中  $i$  跟  $r, c$  都是 **zero-base**) 的數字
    - 注意  $a_i$  我會把他轉乘範圍是  $0 \sim 8$  把數字壓小一點
  - 查詢的部分直接  $a_i = (G/9^i)\%9$
  - 因為若  $9^j < 9^i$  那他就會直接被除掉，那  $9^i < 9^j$  就會被  $\%9$  給 mod 掉
  - 下面就來證明一個比較誇張的例子來證明為什麼會被除掉

proof:  $\sum_{i=0}^k 8 \times 9^i < 1 \times 9^{k+1}$

$$\sum_{i=0}^k 8 \times 9^i$$

$$= 8 \times \frac{9^0 \times (9^{k+1} - 1)}{9 - 1}$$

$$= 8 \times \frac{1 \times (9^{k+1} - 1)}{8}$$

$$= 9^{k+1} - 1$$

$$9^{k+1} - 1 < 9^{k+1} \text{ 故得證}$$

## CODE

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pii pair<int, int>
4 #define pb push_back
5 #define mk make_pair
6 #define F first
7 #define S second
8 #define ALL(x) x.begin(), x.end()
9
```

```

10 using namespace std;
11 using PQ = priority_queue<int, vector<int>, greater<int>>;
12
13 const int INF = 2e18;
14 const int maxn = 3e5 + 5;
15 const int M = 1e9 + 7;
16 const int X = 9;
17
18 int H[11];
19 int n = 3;
20 int G, T;
21 bitset<387420489> dis;
22
23 int swap (int i, int j, int U) {
24     int a = (U / H[i]) % X;
25     int b = (U / H[j]) % X;
26     U += (b - a) * H[i];
27     U += (a - b) * H[j];
28
29     return U;
30 }
31
32 int BFS (int G) {
33     queue<pii> q;
34     q.push ({G, 0}); // 若直接將 G 塞入 map 存 distance 會爆
35     dis[G] = 1;
36
37     while (q.size()) {
38         auto [U, s] = q.front ();
39         q.pop ();
40         if (U == T) return s;
41
42         for (int i = 0; i < 9; i++) {
43             if (i % 3 != 2) {
44                 int C = swap (i, i + 1, U);
45                 if (!dis[C]) {
46                     q.push ({C, s + 1});
47                     dis[C] = 1;
48                 }
49             }
50             if (i < 6) {
51                 int C = swap (i, i + 3, U);
52                 if (!dis[C]) {
53                     q.push ({C, s + 1});
54                     dis[C] = 1;
55                 }
56             }
57         }
58     }
59 }
60
61 void init () {
62     H[0] = 1;
63     for (int i = 1; i <= 9; i++) {
64         H[i] = H[i - 1] * X;

```

```
65     }
66     for (int i = 0; i < 9; i++) {
67         T += i * H[i];
68     }
69     int x;
70     for (int i = 0; i < 9; i++) {
71         cin >> x;
72         x--;
73         G += x * H[i];
74     }
75 }
76
77 void solve () {
78     cout << BFS(G) << "\n";
79 }
80
81 signed main() {
82     // ios::sync_with_stdio(0);
83     // cin.tie(0);
84     int t = 1;
85     //cin >> t;
86     while (t--) {
87         init();
88         solve();
89     }
90 }
```