

D. Fixed Point Guessing

chenjiurierfenjiaohufixedpoint

特殊的一道题目：

- 这里引入了程序交互的背景。可以去学习一下相关的实现

给定一个长度为奇数的数组。 $1 \dots n$;
在这一个数组中, 选定若干组数进行交换。
最终必然有一个位置是不会变化的。
找出这一个数。
采取交互的方式,
有15次询问,
查询区间 $[l, r]$;
 $? \ l \ r$
程序返回该区间上的升序情况。
找到答案后输出! `ans`;

20min

查询偶数长度的区间;
如果当前解在当前区间内, 属于当前区间的数必然是奇数个。
若不是奇数个, 就是另外一半区间二分搜索即可。

一点修正

- 不讨论区间长度的奇偶性, 只需关注这个区间范围内的数的奇偶性。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn = 2e5 + 10;

bool check(int l, int r) //当前的解是否在当前的区间之中。
{
    int t;
    int sum = 0;
    cout << "? " << l << " " << r << "\n";
    fflush(stdout);
    for (int i = l; i <= r; i++)
    {
        cin >> t;
        if (t >= l && t <= r)
            sum++;
    }

    return sum & 1; //奇数还是偶数。在里面就是偶数。
}

void solve()
{
    int n;
```

```

    cin >> n;
    int ans = 0;
    int l = 1, r = n;
    while (l < r) //这里的二分显然不够熟悉。
    {
        int mid = (l + r) >> 1; //向下取整。
        if (check(l, mid)) //如果当前的解在该区间上。
            r = mid;
        else
            l = mid + 1;
    }
    cout << "! " << l << '\n';
    fflush(stdout);
}
int main(){
    int t;
    cin >> t;
    while (t--)
        solve();
}

```

debug的深刻教训

- 不要用关掉缓冲区。否则就会完大蛋。
- 缓冲区这里于flush之间会有一些矛盾。这里接触到了一些与语言方面的细节性的东西。
- 这里的语言点做一个相关的总结。