

2020/07/25 – 2020/07/31周报 week12

本周推荐

by iuiou

- 题

源 <https://codeforces.com/contest/1384/problem/B1> & <https://codeforces.com/contest/1384/problem/B2>

- 题意：一个人要从0号沙滩到 $n+1$ 号岛屿，中间有 n 片海，每片海都有一个深度 d_i 还存在潮汐的作用，潮汐在呈 $[0, 1, 2, \dots, k-1, k, k-1, \dots, 1]$ 的规律按照时间循环变化，在一个时刻海域的深度为潮汐高度加深度，有一个安全深度 l 而且他移动一次需要时间 1 ，它可以在任何一个海域停留任何一个时间。问他能否安全的到达目标岛屿？

- 知识点: dp, 思维, 构造

- 题解

- 题解1：对于简单的版本，考虑使用dp的思想，用 $dp[i][j]$ 表示在时间为 j 时在第 i 个海域的可能性，转移时考虑 $dp[i][j-1]$ 与 $dp[i-1][j-1]$ 有没有存在一个1，有就可以转移，时间总量可以开大一点，开 $2*k*n$ 最后遍历 $dp[n][j]$ 寻找是否存在1即可，至于判断单个状态是否成立，这很简单。
- 题解2：上面那种比较暴力得做法，明显过不了大样例，所以要想一个 $O(n)$ 的想法，其实就是设计一个无论在那种情况都能占优的策略，可以发现，在降潮的时候走是有优势的，如果这时候到 $i+1$ 会有危险，那就等到 $i+1$ 没有危险时再走，二长时间呆在原地，潮水只会往下跌，所以只会越来越“安全”，所以只要找可以等到涨潮的点即可，即满足 $k+d_i \leq l$ 的点，在这个点等到潮水涨到最高，然后再开始继续走，当然，一开始要判断不成立，两种情况，一种是已经涨潮，而对面已经过不去了，还有就是存在一片海域 $d_i > l$ ，那么也一定不成立。时间复杂度 $O(N)$
- 总结：这种题目有时候还是束手无测啊，但是又学到了一种暴力方法dp暴力，感觉很有用啊，其实看见这种一个一个状态划分的非常清楚地题，怎么会想不到dp呢，爬

by QuantumBolt

- 题源 <https://codeforces.com/contest/1384/problem/D>

- 题意 Koa和KoalaKoala两人玩游戏，初始分均为0，每次两人从一个数组中选择一个数，选择后该数字会被从数组中删除，两人的分数异或上该数字的值为新的分数，问均采用最优策略谁能赢，规定KoaKoala先手。

- 知识点：博弈论

- 题解

这个题是博弈论的题：分析一下：设 p_i 表示 n 个数二进制位第 i 位为1的个数 若 $2|p_i$ 那么先手和后手对 i 位的结果没有影响 从而我们需要找到 j 使得 $2 \nmid p_j=1$ 这样才能导致先后手的结果在第 j 位不同 现在我们的任务就是找打最大的 j 位，使结果最大化

1. 若 $\frac{P_j-1}{2}$ 是偶数，那么先手的就一定赢 先手在该位先取一个1，然后跟着后手选，那么后手会选择偶数个1，结果为：先手在该位值为1，后手为0。

2. 若 $\frac{P_{j-1}}{2}$ 是奇数，分两种情况

(1). 若 $2|n = 1$ 先手就必输：先手在该位先取一个1，然后跟着后手选，那么后手会择偶数个1，结果为：先手在该位值为1，后手为0。

(2). 若 $2|n$ 那么先手必赢 先手第一步选一个0，然后将状态转为 $2|n=1$ 的局面，且轮到后手先选择，上面已经证明，这种局面先选择的必输。

团队训练

2020.7.25 牛客多校第六场

2020.7.27 牛客多校第七场

范泽恒

专题

- 无

比赛

- [codeforces round 659\(div2\)](#)
- [Educational Codeforces Round 92 \(Rated for Div. 2\)](#)
- [codeforces round 660\(div2\)](#)

题目

恭天祥

专题

- 无

比赛

- [cf 659 div.2](#)
- [cf 658 div.2](#)

题目

刘怀远

专题

- 无

比赛

- [Codeforces Round 656 \(Div. 3\)](#)

题目

- cf 656 Problem-D:a-Good String
- 链接：<https://codeforc.es/contest/1385/problem/D>
 1. 题意：多组数据，给定长为n的字符串（n为2的整数次方），输出将其变成“a - good string”最少需要改动的字符数。以ch代指字母，“ch - good string”定义为：（1）长为1且字符为ch（2）长大于1，且二分后左边字符全为ch（3）右边为“ch+1 - good string”（4）长大于1，且二分后右边字符全为ch（5）左边为“ch+1 - good string”
 2. 题解：经典二分dfs，维护最小代价即可。

为什么这破题当初能卡那么久

```
#include<bits/stdc++.h>
using namespace std;

typedef long long ll;
const int maxn = 2e5 + 5;
const double pi = acos(-1);
const int mod = 998244353;

int t, n;
char s[131074];

int cal(int l, int r, int c){
    int cnt = 0;
    for(int i = l; i <= r; ++i) if(s[i] != c) ++cnt;
    return cnt;
}

int dfs(int l, int r, char c){
    if(l == r) return s[l] != c;
    int res = 3e7;
    int mid = (l + r) >> 1;
    res = min(res, dfs(l, mid, c + 1) + cal(mid + 1, r, c));
    res = min(res, dfs(mid + 1, r, c + 1) + cal(l, mid, c));
    return res;
}
```

```
int main(){
    scanf("%d", &t);
    while(t--){
        scanf("%d\n", &n);
        scanf("%s", s + 1);
        printf("%d\n", dfs(1, n, 'a'));
    }
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:2020_07_25-2020_07_31%E5%91%A8%E6%8A%A5_week12

Last update: 2020/07/31 17:04