

dp练习2

人员

于家瑞、陶汇笙、洪晨栋、于霄龙、郭栩睿、崔宸赫 到课, 王恩泽 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1583>

#	用户名	姓名	编程分	时间	A	B	C	D	E	F	G
1	yujarui	于家瑞	550	811	100	100	50	100	100	100	
2	yuxiaolong	于霄龙	450	1732	100	100	50	100	100	100	
3	taohuisheng	陶汇笙	350	1711	100	100	50	100		0	
4	hongchenqi	洪晨棋	250	1585	100	100	50				
5	guoxurui	郭栩睿	250	1664	100	100	50	0			
6	hongchendong	洪晨栋	250	1681	100	100	50				
7	wangenze	王恩泽	100	41	0	100					
8	culichenhe	崔宸赫	0	0	0						

您正在共享屏幕 结束共享

本周作业

<https://cppoj.kids123code.com/contest/1684> (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天的 C 题要相对复杂一些, 但是这道题的 $O(n^2)$ 做法是不难的, 这道题如果同学们拿满分觉得比较困难的话, 要求同学们一定要拿到 $O(n^2)$ 的分数

课堂内容

租用游艇 (上周作业)

$f[i]$: 到位置 i 所需要的最少租金

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
int w[maxn][maxn], f[maxn];

int main()
{
    int n; cin >> n;
```

```

for (int i = 1; i <= n; ++i) {
    for (int j = i+1; j <= n; ++j) cin >> w[i][j];
}

memset(f, 0x3f, sizeof(f));
f[1] = 0;
for (int i = 2; i <= n; ++i) {
    for (int j = 1; j <= i-1; ++j) f[i] = min(f[i], f[j] + w[j][i]);
}
cout << f[n] << endl;
return 0;
}

```

HXY和序列

$f[i][j]$: 一共 i 个数, 其中第 i 个数是 j 时, 有多少合法方案

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e3 + 5;
const int mod = 1e9 + 7;
int f[maxn][maxn];
vector<int> vec[maxn];

int main()
{
    for (int i = 1; i < maxn; ++i) {
        for (int j = 1; j <= i; ++j) {
            if (i%j == 0) vec[i].push_back(j);
        }
    }

    int limit, n; cin >> limit >> n;

    f[0][1] = 1;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= limit; ++j) {
            for (int k : vec[j]) {
                if (j%k == 0) f[i][j] = (f[i][j] + f[i-1][k]) % mod;
            }
        }
    }

    int res = 0;
    for (int i = 1; i <= limit; ++i) res = (res + f[n][i]) % mod;
    cout << res << endl;
}

```

```

    return 0;
}

```

最长括号匹配

$f[i]$: 以 i 结尾时, 最长的合法括号子串长度是多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e6 + 5;
char s[maxn];
int f[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);

    int pos = 0;
    for (int i = 1; i <= n; ++i) {
        if (s[i]=='(' || s[i]== '[') continue;

        int len = f[i-1];
        int p = i - len - 1;
        if (s[i] == ')') {
            if (s[p] == '(') f[i] = len+2+f[p-1];
        } else {
            if (s[p] == '[') f[i] = len+2+f[p-1];
        }

        if (f[i] > f[pos]) pos = i;
    }

    int l = pos - f[pos] + 1, r = pos;
    for (int i = l; i <= r; ++i) cout << s[i];
    cout << endl;
    return 0;
}

```

[CSP-J 2025] 异或和 / xor

$f[i]$: 以 i 结尾时, i 可选可不选, 最多能选多少段区间异或值为 k 的子数组出来

2 种转移方式:

- 转移方式一: 不选第 i 个数, $f[i]$ 从 $f[i-1]$ 转移而来

- 转移方式二: 选第 i 个数, 此时, 需要从 i 往前找一个最靠近 i 的位置位置 pos , 要求 $pos \sim i$ 的区间异或值为 k , 此时, $f[i]$ 可以从 $f[pos-1]+1$ 转移而来

暴力实现上述方案的话, 时间复杂度是 $O(n^2)$ 级别的, 上述找位置 pos 可以用 map 来找

map 里存放: 一个前缀异或值所对应的最后一个位置是哪

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int maxn = 5e5 + 5;
int w[maxn], p[maxn], f[maxn];
signed main()
{
    int n, K; cin >> n >> K;
    for (int i = 1; i <= n; ++i) cin >> w[i], p[i] = p[i-1] ^ w[i];

    map<int,int> mp; mp[0] = 0;
    for (int i = 1; i <= n; ++i) {
        int t = p[i] ^ K;
        if (mp.count(t)) {
            int pos = mp[t];
            f[i] = f[pos] + 1;
        }
        mp[p[i]] = i;
        f[i] = max(f[i], f[i-1]);
    }
    cout << f[n] << endl;
    return 0;
}
```