

愉快的代码【8.7】

【7.27】我们特意加入了题解系统，也就是说，姐姐也会重新做一遍给你们题目（包括选做），然后在第二天的题目前给出姐姐自己的代码和注释作为题解或参考

如果觉得自己的代码略为臃肿，可以参考对比一下姐姐的代码；

如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

【7.30】我们特意加入了团队系统，因为感觉到你们有点像是独立学习的样子，比如说姐姐和你们之间有交流，但是你们之间有没有交流呢姐姐就感受不到啦

所以正好在洛谷上发现了一个团队系统，我们可以在这上面布置作业呀（当然姐姐也会继续以 pdf 形式布置作业，你们也还是要以 pdf 形式交作业哈），然后你们就可以在上面看到其它妹妹们的代码呀（包括 AC 代码和还未 AC 的整个过程的代码和分数呀），觉得她们表现不够自己好的话，就可以在群里尽情地嘲笑她们呀～

然后那上面还有一个比赛功能哇，具体形式和我们平时的机考差不多，暑假差不多结束了我们也会有一次期末模拟机考的哈～

如果你们开心的话，你们也可以联合起来给姐姐布置一次平时的作业呀，或者给姐姐安排一次机考呀，你们都是团队的管理员了哈

【8.1】准备给你们留个有趣的团队大作业：给姐姐安排一次机考～

具体时间、题数、难度、知识点待定～

【8.3】经过了某些人性与道德的思考，得出了一个奇怪的想法：

“我今天把代码解决了，明天姐姐的代码还有兴趣看嘛”

那就当天放出来好啦～

同样地：如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

今天的题目：

知识点：DP（一维）

别看少一维，思维上是要比二维要难的哦

二维设 dp 数组的话一般都比较明显

就设个二维数组，代表每个点的状态

而一维的话简单的题也是可以设个一维数组，代表每个点的状态

不过一维的“每个点的状态”不是那么简单设出来的哦～

1、<https://www.luogu.com.cn/problem/P1359>

2、<https://www.luogu.com.cn/problem/P1244>

今天的答案：

8.7问题1：

```
/*
    洛谷P1359：租用游艇
    思想：比较清晰的一维dp。设dp[i]为到达编号为i-1的游艇出租站所需的最少租金。显然转移方程
    为  $dp[i] = \min(dp[0]+a[0][i], dp[1]+a[1][i], \dots, dp[i-1]+a[i-1][i])$ 
    当然这道题也可以用有向图的最短路径来做哦
    时间复杂度：均为 $O(n^2)$ 
*/

// DP
#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int main() {
    int i, j, n;
    cin >> n;
    int a[n][n], dp[n];
    for (i=0; i<n; i++)
        for (j=i+1; j<n; j++) cin >> a[i][j]; // 右上三角矩阵
    for (i=0; i<n; i++) dp[i] = 99999999;
    dp[0] = 0; // 初始状态
    for (i=1; i<n; i++)
        for (j=0; j<i; j++)
            dp[i] = min(dp[i], dp[j]+a[j][i]); // dp
    cout << dp[n-1] << endl;
}

// 最短路 (Dijkstra)
#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int main() {
    int i, j, n;
    cin >> n;
    int a[n][n];
    for (i=0; i<n; i++) { // 构建邻接矩阵
        for (j=0; j<i; j++) a[i][j] = 99999999;
        a[i][i] = 0;
        for (j=i+1; j<n; j++) cin >> a[i][j];
    }
}
```

```

// Dijkstra板子
int vis[n],dij[n],mi,index;
for (i=0; i<n; i++) {
    vis[i] = 0;
    dij[i] = a[0][i];
}
for (i=1; i<n; i++) {
    mi = 99999999;
    for (j=1; j<n; j++)
        if (vis[j] == 0 and dij[j] < mi) {
            mi = dij[j];
            index = j;
        }
    vis[index] = 1;
    for (j=0; j<n; j++)
        dij[j] = min(dij[j],dij[index]+a[index][j]);
}
// 板子结束
cout << dij[n-1] << endl;
}

```

8.7问题2:

```

/*
洛谷P1244：青蛙过河
思想：小小吃思维，设dp[i]为有k个荷叶，i个河心石墩时可以过的青蛙数。我们要让河中心放下尽可能多数量的青蛙，然后下一个青蛙直接跳到右岸石墩上，最后的结果就是那只青蛙的编号。然后开始思维推导：
    当河心石墩数为0时，青蛙只能先走k只在荷叶上，然后k+1号青蛙到右岸石墩，因此右岸石墩底部是k+1，即最多走k+1只(dp[0])青蛙；
    当河心石墩数为1时，由上一步知可以通过荷叶放k+1只(dp[0])青蛙在这个河心石墩上，然后荷叶上放k只，然后下一只青蛙到右岸石墩，因此可以走dp[0]+k+1只；
    当河心石墩数为2时，由上一步知可以通过荷叶和1号河心石墩放dp[1]只青蛙在2号河心石墩上，然后通过荷叶放dp[0]只青蛙在1号河心石墩上，然后荷叶上放k只，然后下一只青蛙到右岸石墩，因此可以走dp[1]+dp[0]+k+1只；
    ...
    当河心石墩数为h时，由上一步知可以通过荷叶和1~(h-1)号河心石墩放dp[h-1]只青蛙在h号河心石墩上，然后通过荷叶和1~(h-2)号河心石墩放dp[h-2]只青蛙在h-1号河心石墩上...然后通过荷叶放dp[0]只青蛙在1号河心石墩上，然后荷叶上放k只，然后下一只青蛙到右岸石墩，因此可以走dp[h-1]+dp[h-2]+...+dp[0]+k+1只；
    时间复杂度：O(n)
    继续进行深入的研究：dp[0] = k+1, dp[1] = dp[0]+k+1 = (k+1)*2, dp[2] = dp[1]+dp[0]+k+1 = (k+1)*4, dp[3] = dp[2]+dp[1]+dp[0]+k+1 = (k+1)*8, ... so?
    时间复杂度：O(1)
*/

// DP
#include <iostream>

```

```

#include <algorithm>
#define ll long long
using namespace std;

int dp[21]; // 需要注意的是，声明全局变量时c++默认值为0，不用初始化
int main() {
    int i,j,h,k;
    cin >> h >> k;
    dp[0] = k+1; // 初始状态
    for (i=1; i<=h; i++) {
        for (j=0; j<i; j++) dp[i] += dp[j]; // dp
        dp[i] += k+1;
    }
    cout << dp[h] << endl;
}

// DP+研究~
#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int main() {
    int h,k;
    cin >> h >> k;
    cout << (k+1)*(1<<h) << endl; // 1<<h = 2^h
}

```

Interesting thing:

<https://www.luogu.com.cn/contest/32821>