

307-07-逐行递推

逐行递推： dp 在某种情况下按照一行一行的顺序进行递推。

P2704 [NOI2001]炮兵阵地

题目描述

司令部的将军们打算在 $N \times M$ 的网格地图上部署他们的炮兵部队。一个 $N \times M$ 的地图由 N 行 M 列组成，地图的每一格可能是山地（用“H”表示），也可能是平原（用“P”表示），如下图。在每一格平原地形上最多可以布置一支炮兵部队（山地上不能够部署炮兵部队）；一支炮兵部队在地图上的攻击范围如图中黑色区域所示：

P	P	H	P	H	H	P	P
P	H	P	H	P	H	P	P
P	P	P	H	H	H	P	H
H	P	H	P	P	P	P	H
H	P	P	P	P	H	P	H
H	P	P	H	P	H	H	P
H	H	H	P	P	P	P	H

如果在地图中的灰色所标识的平原上部署一支炮兵部队，则图中的黑色的网格表示它能够攻击到的区域：沿横向左右各两格，沿纵向上下各两格。图上其它白色网格均攻击不到。从图上可见炮兵的攻击范围不受地形的影响。现在，将军们规划如何部署炮兵部队，在防止误伤的前提下（保证任何两支炮兵部队之间不能互相攻击，即任何一支炮兵部队都不在其他支炮兵部队的攻击范围内），在整个地图区域内最多能够摆放多少我军的炮兵部队。

输入输出格式

输入格式：

第一行包含两个由空格分割开的正整数，分别表示 N 和 M ；

接下来的 N 行，每一行含有连续的 M 个字符（‘P’或者‘H’），中间没有空格。按顺序表示地图中每一行的数据。 $N \leq 100$ ； $M \leq 10$ 。

输出格式：

仅一行，包含一个整数 K ，表示最多能摆放的炮兵部队的数量。

输入样例#1： [复制](#)

5 4
PHPP
PPHH
PPPP
PHPP
PHHP

输出样例#1: 复制

6

题解

这里可以采用逐行递推的方式：定义

$$f(i, s, t) = \max_{0 \leq r < 2^m, s \& r = 0, t \& r = 0, s \& t = 0, (r \gg 1) \& r = 0, (r \gg 2) \& r = 0, \text{map}[i] \& r = 0} \{f(i+1, r, s) + \text{count}[r]\} \quad (1)$$

其中， $f(i, s, t)$ 表示在第*i*行，其前一行的炮兵安排表示为*s*，再前一行的炮兵安排表示为*t*的时候的放炮数量（从第*n*行向第1行转移，其中*r*表示当前这一行的炮兵按放）。由于每一格上，炮兵只能放或者不放，所以可以表示为一个二进制数。

下面，解释max的条件：

1. $0 \leq r < 2^m$ ：穷举所有的当前行的炮兵按放可能性
2. $s \& r = 0$ ：当这一行与上一行没有一列是重复的情况下（取与的操作可以达到这一目的）
3. $t \& r = 0$ ：当这一行与上上行没有一列是重复的情况下
4. $s \& t = 0$ ：当上一行与上上行没有一列是重复的情况下
5. $(r \gg 1) \& r = 0$ ：这一行任意一个炮的相邻位置没炮（位移一位就是相邻）
6. $(r \gg 2) \& r = 0$ ：这一行任意一个炮的相邻2格位置没炮（位移两位就是相邻2格）
7. $\text{map}[i] \& r = 0$ ：这一行的炮兵安排要与地形匹配

$\text{count}[r]$ ：表示*r*这样的安排会有多少门炮（由于是二进制，换言之就是有几个一）。

由于数组会太大，所以需要滚动数组。

AC代码：

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  const int maxn = 1030;
7
8  int n, m, mp[maxn], cnt[maxn], f[2][maxn][maxn];
9
10 vector<int> v;
```



```

59     f[i % 2][s][t] = max(f[i % 2][s][t], f[(i + 1) % 2][r][s] +
cnt[r]);
60     }
61 }
62 }
63 }
64 }
65 // 答案为穷举第二, 第三行所有地形情况下的放炮总数
66 int ans = 0;
67 for (int i = 0; i < v.size(); i++) {
68     int s = v[i];
69     for (int j = 0; j < v.size(); j++) {
70         int t = v[j];
71         ans = max(ans, f[0][s][t]);
72     }
73 }
74 cout << ans << endl;
75 return 0;
76 }

```