

Codeforces Round #662 (Div. 2)

[比赛链接](#)

D. Rarity and New Dress

题意

给定一个 $n \times m$ 的网格，每个网格中有一个字母，问有多少个由同一字母构成的菱形。

下面给出边长为 1, 2, 3 的菱形示例。

```

      a
    a
  a
a
a
  a
    a
      a
    aaa
  aaaa
aaaaa
  aaa
    a

```

题解

考虑以每个网格为中心的菱形个数，发现答案等于以该网格为中心的最大菱形的边长，于是题目等价于求以每个网格为中心得最大菱形的边长的和。

可以将每个菱形划分为左半三角形和右半三角形，再将三角形划分为列，发现每列长度为等差数列。

考虑 dp 求出每个网格所在列能向上拓展的最大长度和向下拓展的最大长度，于是可以得到以每个网格为中心的三角形的列的最大长度。

然后再考虑 dp 根据等差数列的约束即可求出左右三角形的最大边长，最终每个点答案为左右三角形的最大边长的较小值。

```

const int MAXN=2005;
char s[MAXN][MAXN];
int dir[MAXN][MAXN][4], len[MAXN][MAXN];
int main()
{
    int n=read_int(), m=read_int();
    LL ans=0;
    _rep(i, 1, n)
        fgets(s[i]+1, MAXN, stdin);
    _rep(i, 1, n)
        _rep(j, 1, m){
            if(s[i][j]==s[i-1][j])
                dir[i][j][0]=dir[i-1][j][0]+1;
        }
    for(int i=n; i; i--)

```

```
_rep(j,1,m){
    if(s[i][j]==s[i+1][j])
        dir[i][j][1]=dir[i+1][j][1]+1;
    len[i][j]=min(dir[i][j][0],dir[i][j][1]);
}
_rep(i,1,n)
_rep(j,1,m){
    if(s[i][j]==s[i][j-1])
        dir[i][j][2]=min(len[i][j],dir[i][j-1][2]+1);
}
_rep(i,1,n)
for(int j=m;j;j--){
    if(s[i][j]==s[i][j+1])
        dir[i][j][3]=min(len[i][j],dir[i][j+1][3]+1);
    ans+=1+min(dir[i][j][2],dir[i][j][3]);
}
enter(ans);
return 0;
}
```

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_662_div_2

Last update: 2020/08/14 16:47