1296: [SCOI2009]粉刷匠

Time Limit: 10 Sec Memory Limit: 162 MB

Description

windy有 N 条木板需要被粉刷。 每条木板被分为 M 个格子。 每个格子要被刷成红色或蓝色。 windy每次粉刷，只能选择一条木板上一段连续的格子，然后涂上一种颜色。 每个格子最多只能被粉刷一次。 如果windy只能粉刷 T 次，他最多能正确粉刷多少格子？ 一个格子如果未被粉刷或者被粉刷错颜色，就算错误粉刷。

Input

输入文件paint.in第一行包含三个整数，N M T。 接下来有N行，每行一个长度为M的字符串，'0'表示红色，'1'表示蓝色。

Output

输出文件paint.out包含一个整数，最多能正确粉刷的格子数。

Sample Input

3 6 3
111111
000000
001100

Sample Output

16

HINT

30%的数据，满足 1 <= N,M <= 10 ； 0 <= T <= 100 。 100%的数据，满足 1 <= N,M <= 50 ； 0 <= T <= 2500 。

\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\

\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\
\

\
\

没有什么问题是一次DP不能解决的，如果有，那就两次。

```
#include<bits/stdc++.h>
using namespace std;
int n, m, T;
char s[55][55];
int g[55][55][2505], f[55][2505];

inline void putit()
{
    scanf("%d%d%d", &n, &m, &T);
    for(int i = 1; i <= n; ++i) scanf("%s", s[i] + 1);
}

inline int search(int t, int l, int r)
{
    int ret = 0;
    for(int i = l; i <= r; ++i) if(s[t][i] == '0') ret++;
    ret = max(ret, (r - l + 1) - ret);
    return ret;
}

inline void Dp_1()
{
    for(int p = 1; p <= n; ++p)
        for(int i = 1; i <= m; ++i)
            for(int j = 1; j <= i && j <= T; ++j)
                for(int k = 0; k < i; ++k)
                    g[p][i][j] = max(g[p][i][j], g[p][k][j - 1] + search(p, k +
}

inline void Dp_2()
{
    for(int i = 1; i <= n; ++i)
        for(int j = 1; j <= T; ++j)
            for(int k = 0; k <= j; ++k)
                f[i][j] = max(f[i][j], f[i - 1][j - k] + g[i][m][k]);
}

int main()
{
    putit();
    Dp_1();
```

```
    Dp_2();
    printf("%d", f[n][T]);
    return 0;
}
```