

Zuma

一. 考察内容:

动态规划 区间DP

二. 题目分析:

[题目大意]

类似祖玛游戏，有两种颜色的球，0和1，连续三个（或以上）同色球出现在序列中，会使这些球消去，你可以向任意位置添加0或1的球，求消去所有的球最少需要添加多少个球。

[写题思路]

因为处理的问题是无序的，所以考虑区间DP，我们将输入数据进行预处理，我们把连续的同色球看做一个整体，设 $a[i]$ 为第 i 个同色球集合所包含的球个数。

设 $f[i][j]$ 为消去 i 到 j 之间的所有球的最小花费，初始状态： $f[i][i]=\max(0,3-a[i])$ ，结束状态 $f[1][n]$ 。

考虑转移，对于一个区间，消去所有的球有三种方式，

1.将这个区间分为左右两个小区间，分别消去，枚举 k ， $f[i][j]=\min\{f[i][k]+f[k+1][j]\}$

2.将这个区间中间的区间消去之后，两端的球相撞消去， $f[i][j]=f[i+1][j-1]+(a[i]+a[j]<3)$ (i 和 j 需同色)。

3.将这个区间分为两个子区间，这两个子区间中间长度为1的空档，使得分别消去这两个子区间之后，两端的球和中间空档的球相撞消去。枚举 k ， $f[i][j]=\min\{f[i+1][k-1]+f[k+1][j-1]\}$ (i 、 j 、 k 需同色， $a[k]=1, a[i]+a[j]\leq 3$ ，否则有一端的球会提前消去)。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 24
*文件类型: 源代码文件
*题目来源: HDU
*当前状态: 已通过
*备忘录: 区间DP
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 202
int n,a[MAXN],_a,f[MAXN][MAXN],map[MAXN];
/* Variable explain:

*/
void read()
{
    char s[MAXN];
    scanf("%s",s+1);
    n=strlen(s+1);
```

```

    _a=0;
    for(int i=1;i<=n;++i)
        if(s[i]==s[i-1])a[_a]++;
        else a[++_a]=1,map[_a]=s[i];
    return;
}
int dp()
{
    // for(int i=1;i<=_a;++i)printf("%d ",map[i]);puts("");
    for(int i=1;i<=_a;++i)for(int j=1;j<=_a;++j)f[i][j]=1e9;
    for(int i=1;i<=_a;++i)f[i][i]=max(0,3-a[i]);//,printf("f[%d][%d]=%d\n",i,i,f[i][i]);
    for(int i=2;i<=_a;++i)
    {
        for(int j=1;j<=_a-i+1;++j)
        {
            int l=j,r=j+i-1;
            // printf("f[%d][%d]:\n",l,r);
            for(int k=l;k<r;++k)f[l][r]=min(f[l][r],f[l][k]+f[k+1]
[r]);//,printf("f[%d][%d]+[%d][%d]\n",l,k,k+1,r);
            if(map[l]==map[r])f[l][r]=min(f[l][r],f[l+1][r-1]+(a[l]
+a[r]<3));//,printf("f[%d][%d]+%d\n",l+1,r-1,a[l]+a[r]<3);
            for(int k=l+1;k<r;++k)
                if(map[l]==map[k]&&map[l]==map[r]&&a[k]==1&&a[l]+a[r]<=3)
                    f[l][r]=min(f[l][r],f[l+1][k-1]+f[k+1][r-1]);//,printf("f[%d]
[%d]+f[%d][%d]\n",l+1,k-1,k+1,r-1);
            // printf("f[%d][%d]=%d\n",l,r,f[l][r]);
        }
    }
    return f[1][_a];
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    int T;
    scanf("%d",&T);
    for(int i=0;i<T;++i)
    {
        read();
        printf("Case #d: %d\n",i+1,dp());
    }
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)