

3618: 剪刀石头布

题目描述

你可能听说过“剪刀，石头，布”这个游戏。规则很简单。两人数到三，然后同时做一个表示石头、剪刀或布的手势。石头打剪刀，剪刀打布，布打石头。例如，如果第一个人做“石头”手势，第二个人做“布”手势，则第二个人获胜。当然，如果两个人做相同的手势，就打平。TT想在“剪刀，石头，布”（ $1 \leq n \leq 100000$ ）的N场比赛中与他的好朋友YY比赛。YY是这项运动的专家，他能在TT做出动作之前预测出每一个动作。不幸的是，YY非常懒惰。因此，她倾向于连续多次地做同样的手势。事实上，她只愿意在整个游戏中最多切换一次手势。例如，她可能会在前X个游戏中出“石头”，然后在剩余的N-X游戏中切换到“布”。考虑到TT将要玩的手势顺序，请确定YY能赢的最大游戏数。

输入

第一行包含n。剩下的n行包含TT的手势，每个手势都是S、H或P（分别表示剪刀，石头和布）。

输出

输出YY能赢的最大游戏数，她最多只能更改一次手势。

样例输入

```
1 | 5
2 | P
3 | P
4 | H
5 | P
6 | S
```

样例输出

```
1 | 4
```

##

思路

//钱嘉欢

```
n=5
i= 1 2 3 4 5 6
s={0, 0, 0, 0, 0, 1}
h={0, 0, 0, 1, 1, 1}
p={0, 1, 2, 2, 3, 3}
```

//严老师的话

```

1   if(ans<s[i]+h[n]-h[i]) ans=s[i]+h[n]-h[i]; 前面剪刀的个数（就是得分点），后面石头的个数（也是
   一个得分点）
2   if(ans<s[i]+p[n]-p[i]) ans=s[i]+p[n]-p[i];前面剪刀的个数，后面布的个数
3   if(ans<p[i]+h[n]-h[i]) ans=p[i]+h[n]-h[i];前面布的个数，后面石头的个数
4   if(ans<p[i]+s[n]-s[i]) ans=p[i]+s[n]-s[i];前面布的个数，后面剪刀的个数
5   if(ans<h[i]+p[n]-p[i]) ans=h[i]+p[n]-p[i];前面石头的个数，后面布的个数
6   if(ans<h[i]+s[n]-s[i]) ans=h[i]+s[n]-s[i];前面石头的个数，后面剪刀的个数

```

代码

```

1  //钱嘉欢
2  #include<iostream>
3  using namespace std;
4  long long s[100005],h[100005],p[100005],ans,n;
5  char x;
6  int main()
7  {
8      cin>>n;//输入一共要进行几局
9      for(int i=1;i<=n;i++)
10     {
11         cin>>x;//石头剪刀布中YY出的一个
12         if(x=='s') s[i]=1;//如果YY出剪刀，s[i]就等于1
13         if(x=='h') h[i]=1;//如果YY出石头，h[i]就等于1
14         if(x=='p') p[i]=1;//如果YY出布，p[i]就等于1
15         s[i]+=s[i-1];//s[i]再加上前面一个
16         h[i]+=h[i-1];//h[i]再加上前面一个
17         p[i]+=p[i-1];//p[i]再加上前面一个
18     }
19     //看图跟容易理解
20     for(int i=0;i<=n+1;i++)
21     {
22         //ans存的是答案
23         if(ans<s[i]+h[n]-h[i]) ans=s[i]+h[n]-h[i];
24         if(ans<s[i]+p[n]-p[i]) ans=s[i]+p[n]-p[i];
25         if(ans<p[i]+h[n]-h[i]) ans=p[i]+h[n]-h[i];
26         if(ans<p[i]+s[n]-s[i]) ans=p[i]+s[n]-s[i];
27         if(ans<h[i]+p[n]-p[i]) ans=h[i]+p[n]-p[i];
28         if(ans<h[i]+s[n]-s[i]) ans=h[i]+s[n]-s[i];
29         //一个一个列出来，然后在一个一个的比较，如果比ans大，ans就是那个数
30     }
31     cout<<ans;//输出
32     return 0;
33 }
34

```

思路

//吴佳轩（四年级）

因为题目中描述的话，他里面手势可以切换一次，那么万一他最大的值，比另外两两个加起来，因为它可以切换一次，那么后面两次加起来的小的话，那么它的最大值就应该变成后面两个加起来的值，所以说要进行那个要进行比较，然后才能判断他是不是那个比他大的或比她小的。

代码

```

1 //吴佳轩
2 #include<bits/stdc++.h>
3 using namespace std;
4 int h[100005],p[100005],s[100005];
5 int main()
6 {
7     int n;
8     cin>>n;
9     int sumh,sump,sums;
10    sumh=sump=sums=0;
11    int maxn=0;
12    char c;
13    for(int i=0;i<n;i++)
14    {
15        cin>>c;
16        if(c=='P')
17        {
18            sump++;//如果c等于布，那么布的计数器++
19        }
20        if(c=='H')
21        {
22            sumh++;//如果c等于石头，那么石头的计数器++
23        }
24        if(c=='S')
25        {
26            sums++;//如果c等于剪刀，那么剪刀的计数器++
27        }
28        h[i]=sumh;//把计数器的值赋值给数组
29        p[i]=sump;//把计数器的值赋值给数组
30        s[i]=sums;//把计数器的值赋值给数组
31    }
32    if(maxn<sums)
33        maxn=sums;
34    if(maxn<sump)
35        maxn=sump;
36    if(maxn<sumh)
37        maxn=sumh;//以上求出最大值
38    for(int i=0;i<n;i++)
39    {
40        if(maxn<h[i]+sump-p[i])
41            maxn=h[i]+sump-p[i];//2个计数器的值 和最大值进行比较。
42        if(maxn<p[i]+sumh-h[i])
43            maxn=p[i]+sumh-h[i];//2个计数器的值 和最大值进行比较。
44        if(maxn<h[i]+sums-s[i])
45            maxn=h[i]+sums-s[i];//2个计数器的值 和最大值进行比较。
46        if(maxn<s[i]+sumh-h[i])
47            maxn=s[i]+sumh-h[i];//2个计数器的值 和最大值进行比较。
48        if(maxn<p[i]+sums-s[i])
49            maxn=p[i]+sums-s[i];//2个计数器的值 和最大值进行比较。
50        if(maxn<s[i]+sump-p[i])
51            maxn=s[i]+sump-p[i];//2个计数器的值 和最大值进行比较。
52    }
53    cout<<maxn;
54    return 0;
55 }

```

思路

李明烨思路：用区间和枚举每个比赛的最好情况，然后“皇帝”问他的6个大臣答案，然后和自己共7个答案决策，

严老师的话，其实和我说明说的思路一致。

代码

```
1 #include<cstdio>
2 #include<cstring>
3 #include<iostream>
4 #include<algorithm>
5 using namespace std;
6 long long n/*比赛场次*/,win/*最多能赢场次*/,s/*剪刀*/,h/*石头*/,p/*布*/,cs[100005]/*剪刀区间和
  */ ,ch[100005]/*石头区间和*/,cp[100005]/*布区间和*/;
7 char c;//出什么
8 int main(){
9     scanf("%lld\n",&n);//输入
10    for(int i=1;i<=n;i++){
11        scanf("%c",&c);
12        getchar();
13        switch(c){
14            case 'S':s++;break;//是剪刀就加一个
15            case 'H':h++;break;//是石头就加一个
16            case 'P':p++;break;//是布就加一个
17        }
18        cs[i]=s;ch[i]=h;cp[i]=p;//记录区间和
19    }//输入
20    win=max(s,max(h,p));//如果不改，能赢几局（剪刀石头布中最多的）
21    for(int i=1;i<=n-1;i++){//枚举所有的断点（相当于题目中的“x”）
22        win=max(win,max(cs[i]+h-ch[i],max(cs[i]+p-cp[i],max(ch[i]+s-cs[i],max(ch[i]+p-
  cp[i],max(cp[i]+s-cs[i],cp[i]+h-ch[i]))))));//6个点枚举最大的（超懒打法！）
23    }
24    printf("%lld\n",win);//输出
25    return 0;
26 }
```

```
1 //殷学楷
2 #include<bits/stdc++.h>
3 using namespace std;
4 char c;
5 int n,a[100005],x[5]={0,2,5},maxx,sum1[10],sum2[10];
6 int main() {
7     cin>>n;
8     for(int i=1; i<=n; i++) {
9         cin>>c;
10        if(c=='P') a[i]=5;
11        if(c=='H') a[i]=0;
12        if(c=='S') a[i]=2;
13        sum1[a[i]]++;
14        getchar();
15    }
16    for(int k=1; k<=n; k++) {
17        sum2[a[k]]++;
18        for(int j=0; j<3; j++) {
19            for(int j1=0; j1<3; j1++) {
20                int cnt=0;
21                if(x[j1]==0) cnt=cnt+sum1[2]-sum2[2];
22                if(x[j1]==2) cnt=cnt+sum1[5]-sum2[5];
23                if(x[j1]==5) cnt=cnt+sum1[0]-sum2[0];
```

```
24         if(x[j]==0) cnt+=sum2[2];
25         if(x[j]==2) cnt+=sum2[5];
26         if(x[j]==5) cnt+=sum2[0];
27         if(maxx<cnt) maxx=cnt;
28     }
29 }
30 }
31 cout<<maxx<<endl;
32 return 0;
33 }
```