Manacher算法

1.使用Manacher算法求最长回文子串

题目：给出一个字符串，计算出其中的最长回文子串的长度。

如果遍历每一个字符，并以该字符为中心向两边查找，时间复杂度为$O(n^2)$

Manacher算法，也叫马拉车算法，可以在O(n)下解决。

Manacher算法原理

> *对于p[i]，如果i，设j是i关于id*对称点，如图所示，则基于以下三种情况，可以求出p[i]的值：*
>
> - *（1）以j为中心的回文串有一部分在以id为中心的回文串之外。因为mx是以id为中心的最长回文的右边界，所以以i为中心的回文串不可能会有字符在以id为中心的回文串之外；否则mx就不是以id为中心的最长回文的右边界。所以，在这种情况下，p[i]=mx−i。*
> - *（2）以j为中心的回文串全部在以id为中心的回文串的内部，则p[i]=p[j]，而且p[i]不可能再增加。*
> - *（3）以j为中心的回文串的左端正好与以id为中心的回文串的左端重合。则p[i]=p[j]或p[i]=mx−i，并且p[i]还有可能会继续增加，即while (s_new[i-p[i]]==s_new[i+p[i]]) p[i]++;*
>
> *所以，if (i < mx) p[i] = min(p[2 * id - i], mx- i)；其中2*id - i为i关于id的对称点，即上面的j点，而p[j]表示以j为中心的最长回文半径，因此可以利用p[j]来加快求解p[i]。*

模板题：POJ3974

HDJ 3613

```
// poj 3974
#include "iostream"
```

```cpp
#include "cstring"
using namespace  std;
const int MAXN=1E6+10;
char ss[MAXN],ss_new[MAXN*2+6];
int p[MAXN*2+6];
int len_ss;
void init_ss()
{
    len_ss=strlen(ss);
    ss_new[0]='$';
    for(int i=1;i<=len_ss;++i){
        ss_new[i*2-1]='#';
        ss_new[i*2]=ss[i-1];
    }
    ss_new[len_ss*2+1]='#';
    ss_new[len_ss*2+2]='@';
}
void Manacher() {
    int rightRange = 0, center, ans = 0;
    memset(p, 0, sizeof(p));
    for (int i = 1; i <= len_ss * 2 + 1; ++i)
    {
        if(rightRange>i)p[i]=min(rightRange-i,p[center*2-i]);
        else p[i]=1;
        while (ss_new[i+p[i]]==ss_new[i-p[i]])++p[i];

        if(p[i]+i>rightRange)
            rightRange=p[i]+i,center=i;
    ans=max(ans,p[i]);
    }
    printf("%d\n",ans-1);
}
int main()
{
//    freopen("in.text","r",stdin);
    int cas=0;
    while (scanf("%s",ss))
    {
        if(!strcmp(ss,"END"))break;
        init_ss();
        printf("Case %d: ",++cas);
        Manacher();
    }
}

```