字符串其他技巧

马拉车

问题描述

给定一个长度为n的字符串s，请找到所有对(i,j)使得子串$s[i...j]$为一个回文串。

给定一个长度为n的字符串s，请找到最长回文子串及其长度

解法

①枚举中心点以及回文串长度，在$O(n^2)$复杂度下解决问题。

②字符串哈希，复杂度$O(nlogn)$

③后缀数组和快速LCA，复杂度$O(n)$

④Manacher在解决这类问题相等简单，复杂度$O(n)$且常数更小。

Manacher 算法流程

我们用$d_1[i]$和$d_2[i]$分别表示以位置i为中心的长度为奇数和长度为偶数的回文串个数。

我们讨论寻找所有奇数长度的子回文串的情况。

维护已找到的最靠右的子回文串的边界(l,r),初始时设l=0,r=-1。

我们要对下一个i进行转移，之前所有的$d_1[i]$都已经计算完毕，通过下列方式进行计算：

①当i处于当前子回文串外，即$i>r$时时，用朴素算法；

②当$i\le r$时，通过已计算的$d_1[]$中获得信息。

（回文串中左右两边的d_i是相同的，都可以有$d_1[i]=d_1[j]$）

当内部的回文串达到外部的边界，即$j-d_1[j]+1\le l$时，外部的对称性没有保证。

要先令$d_1[i]=r-i$，用朴素算法尽可能增加$d_1[i]$的值。

统一处理

我们可以用一个小技巧将d1和d2的计算统一。

比如给定字符串s=ababc，n=5，我们可以转化为长度k=2*n+1的字符串s'=#a#b#a#b#c#。

在s中以一个字母为中心的长度为m+1的极大子回文串，对应在s'中的以相同字母为中心，长度为2m+3的极大子回文串；而s中一个以空为中心长度为m的极大子回文串，对应于s'中对应一个#为中心，长度为2m+1的极大子回文串。

简化后的模板

```
void getstr() {//重定义字符串
    int k = 0;
    str[k++] = '@';//开头加个特殊字符防止越界
    for (int i = 0; i < len; i++) {
        str[k++] = '#';
        str[k++] = s[i];
    }
    str[k++] = '#';
    len = k;
    str[k] = 0;//字符串尾设置为0，防止越界
}

int manacher() {
    int mx = 0, id;//mx为最右边，id为中心点
    int maxx = 0;
    for (int i = 1; i < len; i++) {
        if (mx > i) Len[i] = min(mx - i, Len[2 * id - i]);//判断当前点超没超过mx
        else Len[i] = 1;//超过了就让他等于1，之后再进行查找
        while (str[i + Len[i]] == str[i - Len[i]]) Len[i]++;//判断当前点是不是最长回
文子串，不断的向右扩展
        if (Len[i] + i > mx) {//更新mx
            mx = Len[i] + i;
            id = i;//更新中间点
            maxx = max(maxx, Len[i]);//最长回文字串长度
        }
    }
    return (maxx - 1);
}
```

哈希找最长回文子串

通过哈希同样可以$O(n)$解决这个问题，具体方法就是记R_i表示以i作为结尾的最长回文的长度，那么答案就是$max_{i=1}^{n}R_i$。考虑到$R_i \leq R_{i-1}+2$，因此我们只需要暴力从$R_{i-1}+2$开始递减，直到找到第一个回文即可。记变量z表示当前枚举的R_i，初始时为0，则 在每次z增大的时候都会增大2，之后每次暴力循环都会减少1，故暴力循环最多发生2n次，总的时间复杂度为$O(n)$。

最小表示法

可以用$O(n)$的复杂度得到字典序最小/最大的字符串旋转结果。

```
int get_minstring(char *s){
    int len=strlen(s);
    int i=0,j=1,k=0;
    while(i<len&&j<len&&k<len){
```

```
        int t=s[(i+k)%len]-s[(j+k)%len];
        if(t==0) k++;
        else{
            if(t>0) i+=k+1;
            else j+=k+1;
            if(i==j) j++;
            k=0;
        }
    }
    return min(i,j);
}

int get_maxstring(char *s){
    int len=strlen(s);
    int i=0,j=1,k=0;
    while(i<len&&j<len&&k<len){
        int t=s[(i+k)%len]-s[(j+k)%len];
        if(t==0) k++;
        else{
            if(t>0) j+=k+1;
            else i+=k+1;
            if(i==j) j++;
            k=0;
        }
    }
    return min(i,j);
}
```

例题

luoguP4555 国家集训队]最长双回文串(马拉车)

```
#include<bits/stdc++.h>
using namespace std;
const int N=3e5+7;
const int mod=1e9+7;
string str;
char s[N];
int k,ans=0,Len[N],ll[N],rr[N];


void Manacher(){
    int r=0,id,mx=0;
    for(int i=1;i<=k;i++){
        if(r>i) Len[i]=min(r-i,Len[2*id-i]);
        else Len[i]=1;
        while(s[i+Len[i]]==s[i-Len[i]]) Len[i]++;
        ll[i+Len[i]-1]=max(Len[i]-1,ll[i+Len[i]-1]);
        rr[i-Len[i]+1]=max(Len[i]-1,rr[i-Len[i]+1]);
        if(Len[i]+i>r){
            r=Len[i]+i;
            id=i;
        }
    }
}
```

```
signed main(){
    cin>>str;
    k=0;
    s[0]='@';
    s[++k]='#';
    for(int i=0;i<str.size();i++){
        s[++k]=str[i];
        s[++k]='#';
    }
    s[++k]='\0';
    Manacher();
    for(int i=1;i<=k;i+=2) rr[i]=max(rr[i],rr[i-2]-2);
    for(int i=k-1;i>=1;i-=2) ll[i]=max(ll[i],ll[i+2]-2);
    for(int i=1;i<=k;i+=2) if(ll[i]&&rr[i]) ans=max(ans,ll[i]+rr[i]);
    cout<<ans<<"\n";
    return 0;
}
```

luoguP1659 [国家集训队]拉拉队排练(马拉车)

求所有奇长回文串，并将长度连乘作为答案。

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int N=2e6+7;
const int mod=19930726;

int n,k,x,Len[N],cnt[N];
long long ans=1,maxx=0;
char str[N];
priority_queue<int,vector<int>,less<int> >q;

inline int fpow(int a,int b){
    int res=1;
    while(b){
        if(b&1) res=res*a%mod;
        a=a*a%mod;
        b>>=1;
    }
    return res;
}

void Manacher(){
    int mx=0,id;
    for(int i=1;i<=n;i++){
        if(mx>i) Len[i]=min(mx-i,Len[2*id-i]);
        else Len[i]=1;
        while(str[i+Len[i]]==str[i-Len[i]]) Len[i]++;
        if(Len[i]+i>mx){
            maxx=max(maxx,Len[i]*2-1);
            mx=Len[i]+i;
            id=i;
        }
```

```
        cnt[Len[i]*2-1]++;
    }
}

signed main(){
    scanf("%lld%lld",&n,&k);
    scanf("%s",str+1);
    str[0]='@';str[n+1]='&';
    Manacher();
    ans=1;
    for(int i=maxx;i>=1;i--){
        cnt[i]+=cnt[i+2];
        if(k>cnt[i]){
            ans=ans*fpow(i,cnt[i])%mod;
            k-=cnt[i];
        }else{
            ans=ans*fpow(i,k)%mod;
            k=0;
            break;
        }
    }
    if(k) puts("-1");
    else printf("%lld",ans);
    return 0;
}
```

参考资料

OI-wiki

https://blog.csdn.net/qq_43456058/article/details/94588721