POJ 3261 Milk Patterns Solution

write by Gestalti Lur
2012-08-08

题目大意

求字符串 S 中出现不小于 K 次的子串中最长的长度。

算法分析

把 height 数组分段并二分长度 L，每一段中的 height 都是大于 L 的，如果这一段 height 的个数不小于 K，那么 L 就是可行的。

参考代码

```
/*
  POJ 3261
  DC3 TEST
  ACCEPTED
  2012-08-03
*/
#include<cstdio>
#include<cstring>
#define MAXN 20013

int wa[MAXN],wb[MAXN],wv[MAXN],ws[ 1000002 ];

inline int F( int x , int tb )
{ return ((x)/3+((x)%3==1?0:tb)); }

inline int G( int x , int tb )
{ return ((x)<tb?(x)*3+1:((x)-tb)*3+2 ); }

int c0(int *r,int a,int b)
{return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}
int c12(int k,int *r,int a,int b)
{if(k==2) return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);
 else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];}

void sort(int *r,int *a,int *b,int n,int m)
{
    int i;
    for(i=0;i<n;i++) wv[i]=r[a[i]];
    for(i=0;i<m;i++) ws[i]=0;
    for(i=0;i<n;i++) ws[wv[i]]++;
    for(i=1;i<m;i++) ws[i]+=ws[i-1];
```

```
    for(i=n-1;i>=0;i--) b[--ws[wv[i]]]=a[i];
    return;
}
void dc3(int *r,int *sa,int n,int m)
{
    int i,j,*rn=r+n,*san=sa+n,ta=0,tb=(n+1)/3,tbc=0,p;
    r[n]=r[n+1]=0;
    for(i=0;i<n;i++) if(i%3!=0) wa[tbc++]=i;
    sort(r+2,wa,wb,tbc,m);
    sort(r+1,wb,wa,tbc,m);
    sort(r,wa,wb,tbc,m);
    for(p=1,rn[F(wb[0] , tb )]=0,i=1;i<tbc;i++)
      rn[F(wb[i] , tb )]=c0(r,wb[i-1],wb[i])?p-1:p++;
    if(p<tbc) dc3(rn,san,tbc,p);
    else for(i=0;i<tbc;i++) san[rn[i]]=i;
    for(i=0;i<tbc;i++) if(san[i]<tb) wb[ta++]=san[i]*3;
    if(n%3==1) wb[ta++]=n-1;
    sort(r,wb,wa,ta,m);
    for(i=0;i<tbc;i++) wv[ wb[ i ] = G( san[i] , tb )]= i;
    for(i=0,j=0,p=0;i<ta && j<tbc;p++)
    sa[p]=c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];
    for(;i<ta;p++) sa[p]=wa[i++];
    for(;j<tbc;p++) sa[p]=wb[j++];
    return;
}

int n , m , len;
int r[MAXN * 3 ] , sa[MAXN * 3 ];
int rank[MAXN],h[MAXN];
void calheight()
{
    int i,j,k=0;
    for(i=1;i<=n;i++) rank[sa[i]]=i;
    for(i=0;i<n;h[rank[i++]]=k)
    for(k?k--:0,j=sa[rank[i]-1];r[i+k]==r[j+k];k++);
    return;
}

char str[ MAXN ];
void init()
{
  m = 0;
  for( int i = 0 ; i < n ; ++ i )
    {
     scanf( "%d" , &r[ i ] );
     m = m > ( r[ i ] + 1 ) ? m : ( r[ i ] + 1 );
    }
  r[ n ] = 0;
```

```
  return ;
}

int ans;
inline bool find( int k )
{
  int i = 0 , j , cnt = 0;
  while( i < n )
    {
     j = i + 1;
     if( h[ j ] >= k )
     {
       cnt = 1;
       while( j <= n and h[ j ] >= k )
         {
           ++ cnt;
           ++ j;
         }
       if( cnt >= len )
         {
           ans = k;
           return true;
         }
     }
     i = j;
    }
  return false;
}

void bipart()
{
  //for( int i = 0 ; i <= n ; ++ i ) printf( "%d " , h[ i ] ); printf( "\n" );
  int ls , rs , mid;
  ans = 0;
  ls = 0 , rs = n;
  while( ls < rs )
    {
     mid = ls + rs >> 1;
     if( find( mid ) )
     ls = mid + 1;
     else
     rs = mid;
    }
  printf( "%d\n" , ans );
  return ;
}

int main()
```

```
{
 while( scanf( "%d%d" , &n , &len ) not_eq EOF )
   {
    init();
    dc3( r , sa , n + 1 , m );
    calheight();
    bipart();
   }
 return 0;
}
```