P3369 【模板】普通平衡树

```
int ch[N][2],size[N],cnt[N],cntN,root,par[N],val[N],n;

inline int chk(ci x){
   return ch[par[x]][1]==x;
}

inline void pushup(ci x){
   size[x]=size[ch[x][0]]+size[ch[x][1]]+cnt[x];
}

inline void rotate(int x){
   int y=par[x],z=par[y],k=chk(x),w=ch[x][k^1];
   ch[y][k]=w;
   par[w]=y;
   ch[z][chk(y)]=x;
   par[x]=z;
   ch[x][k^1]=y;
   par[y]=x;
   pushup(y);
   pushup(x);
}

inline void splay(int x,int goal=0){
   while(par[x]!=goal){
      int y=par[x],z=par[y];
      if(z!=goal){
         if(chk(x)==chk(y)){
            rotate(y);
         }
         else rotate(x);
      }
      rotate(x);
   }
   if(!goal)root=x;
}

inline void find(int x){
   int cur=root;
   while(val[cur]!=x&&ch[cur][x>val[cur]])cur=ch[cur][x>val[cur]];
   splay(cur);
}

inline void insert(int x){
```

```
    int cur=root,p=0;
    while(val[cur]!=x&&cur){
        p=cur;
        cur=ch[cur][x>val[cur]];
    }
    if(cur)++cnt[cur];
    else{
        cur=++cntN;
        if(p)ch[p][x>val[p]]=cur;
        par[cur]=p;
        ch[cur][0]=ch[cur][1]=0;
        size[cur]=cnt[cur]=1;
        val[cur]=x;
    }
    splay(cur);
}

inline int pre(int x){
    find(x);
    if(val[root]<x)return root;
    int cur=ch[root][0];
    while(ch[cur][1])cur=ch[cur][1];
    return cur;
}

inline int suc(int x){
    find(x);
    if(val[root]>x)return root;
    int cur=ch[root][1];
    while(ch[cur][0])cur=ch[cur][0];
    return cur;
}

inline int kth(int x){//返回的是第 k 大的节点，不是值
    ++x;int cur=root;
    while(1){
        if(size[ch[cur][0]]>=x)cur=ch[cur][0];
        else if(size[ch[cur][0]]+cnt[cur]>=x)return cur;
        else{
            x-=size[ch[cur][0]]+cnt[cur];
            cur=ch[cur][1];
        }
    }
}

inline int rank(int x){
    find(x);
    return size[ch[root][0]];
}
```

```
        inline void Del(int x){
           int lst=pre(x),nxt=suc(x);
           splay(lst);
           splay(nxt,lst);
           int del=ch[nxt][0];
           if(cnt[del]>1){
              --cnt[del];
              splay(del);
           }
           else ch[nxt][0]=0;
        }

        //#undef int
        int main()
        {
        //#define int long long
            freopen("read.txt","r",stdin);
           //freopen("write.txt","w",stdout);
           rd(n);int op,x;
           insert(inf);insert(-inf);
           rep(i,1,n){
              rd(op),rd(x);
              if(op==1)insert(x);
              else if(op==2)Del(x);
              else if(op==3)printf("%d\n",rank(x));
              else if(op==4)printf("%d\n",val[kth(x)]);
              else if(op==5)printf("%d\n",val[pre(x)]);
              else printf("%d\n",val[suc(x)]);
           }
           //printf("\n 内存消耗: %.3lf M",(double)sizeof(ch)/(1<<20));
           return 0;
        }
```

P3834 【模板】可持久化线段树 1（主席树）

```
int a[N],n,m,cnt,b[N],id[N],rt[N],un;
struct cmp{
   inline bool operator()(const int&T1,const int&T2){
      return T1<T2;
   }
};
struct TREE{
   int l,r,sum;
}t[N*25];

void buildtree(int old,int &now,int pos,int l,int r){
   t[++cnt]=t[old];now=cnt;++t[now].sum;
   if(l==r)return;
   prepare;
   if(pos<=mid)buildtree(t[old].l,t[now].l,pos,l,mid);
```

```
    if(pos>mid)buildtree(t[old].r,t[now].r,pos,mid+1,r);
}

int query(int left,int right,int k,int l,int r){//left,right 代表查询区间的左右端点
    if(l==r)return l;
    prepare;
    int tsum=t[t[right].l].sum-t[t[left].l].sum;//注意！！！
    if(k<=tsum)return query(t[left].l,t[right].l,k,l,mid);
    if(k>tsum)return query(t[left].r,t[right].r,k-tsum,mid+1,r);
}

//#undef int
int main()
{
//#define int long long
    freopen("read.txt","r",stdin);
    //freopen("write1.txt","w",stdout);
    rd(n),rd(m);
    rep(i,1,n){
        rd(a[i]);
        b[i]=a[i];
    }
    sort(b+1,b+n+1,cmp());
    un=unique(b+1,b+n+1)-b-1;
    rep(i,1,n)id[i]=lower_bound(b+1,b+un+1,a[i])-b;
    rep(i,1,n)buildtree(rt[i-1],rt[i],id[i],1,un);
    int x,y,k;
    rep(i,1,m){
        rd(x),rd(y),rd(k);
        printf("%d\n",b[query(rt[x-1],rt[y],k,1,un)]);
    }
    //printf("\n 内存消耗: %.3lf M",(double)sizeof(a)/(1<<20));
    return 0;
}
```

P3805 【模板】manacher 算法

```
char s[N],str[N<<1];
int ans,cnt,r[N<<1];

//#undef int
int main()
{
//#define int long long
    //freopen("read.txt","r",stdin);
    scanf("%s",s+1);
    int len=strlen(s+1);
    //预处理
    str[0]='%';
    rep(i,1,len){
```

```
        str[++cnt]='#';
        str[++cnt]=s[i];
    }
    str[++cnt]='#';
    len=cnt;int mx=0,id;
    rep(i,1,len){
        if(i<=mx)r[i]=min(mx-i+1,r[(id<<1)-i]);
        else r[i]=1;
        while(str[i-r[i]]==str[i+r[i]])++r[i];
        if(i+r[i]-1>mx){
            mx=i+r[i]-1;
            id=i;
            ans=max(ans,r[i]);
        }
    }
    printf("%d\n",ans-1);
    //printf("内存消耗：%.3f M",(double)sizeof(str)/(1<<20));
    return 0;
}
```

P3389 【模板】高斯消元法

```
int n;
double a[N][N];

//#undef int
int main()
{
//#define int long long
    //freopen("read.txt","r",stdin);
    scanf("%d",&n);
    rep(i,1,n){
       rep(j,1,n+1){
          scanf("%lf",&a[i][j]);
        }
    }
    rep(i,1,n){
        int r=i;
        rep(j,i+1,n){
            if(fabs(a[j][i])>fabs(a[r][i]))r=j;
        }
        if(fabs(a[r][i])<Eps){
            puts("No Solution");
            return 0;
        }
        if(r!=i)swap(a[i],a[r]);
        rep(j,i+1,n){
            double div=a[j][i]/a[i][i];
            rep(k,i,n+1){
                a[j][k]-=div*a[i][k];
```

```
            }
        }
    }
    dwn(i,n,1){
        rep(j,i+1,n){
            a[i][n+1]-=a[j][n+1]*a[i][j];
        }
        a[i][n+1]/=a[i][i];
    }
    rep(i,1,n)printf("%.2lf\n",a[i][n+1]);
    //printf("\n 内存消耗：%.3f M",(double)sizeof(a)/(1<<20));
    return 0;
}
```

P1452 Beauty Contest（【模板】旋转卡壳，内含二维凸包）

```
int n,top,stk[N];
int ans=0;
struct VEC{
    double x,y;
    VEC(double x1=0.0,double y1=0.0){
        x=x1;y=y1;
    }
}a[N];

inline int dcmp(const double&x){
    if(fabs(x)<Eps)return 0;
    return x<0.0?-1:1;
}

inline double X(const VEC&T0,const VEC&T1,const VEC&T2){
    return (T1.x-T0.x)*(T2.y-T0.y)-(T2.x-T0.x)*(T1.y-T0.y);
}

inline int dist(const VEC&T1,const VEC&T2){
    return (int)(T1.x-T2.x)*(int)(T1.x-T2.x)+(int)(T1.y-T2.y)*(int)(T1.y-T2.y);
}

inline double lenf(const VEC&T1){
    return T1.x*T1.x+T1.y*T1.y;
}

inline bool cmp(const VEC&T1,const VEC&T2){
    double res=X(a[1],T1,T2);
    if(dcmp(res)>0)return true;
    if(dcmp(res)<0)return false;
    return lenf(T1)<lenf(T2);
```

```
}


//#undef int
int main()
{
//#define int long long
    //freopen("testdata (3).in","r",stdin);
    scanf("%d",&n);
    rep(i,1,n){
       scanf("%lf%lf",&a[i].x,&a[i].y);
    }
    rep(i,2,n){
        if(a[i].y<a[1].y||(dcmp(a[i].y-a[1].y)==0&&a[i].x<a[1].x))swap(a[1],a[i]);
    }
    sort(a+2,a+n+1,cmp);
    stk[++top]=1;stk[++top]=2;
    rep(i,3,n){
        while(top>1&&dcmp(X(a[stk[top-1]],a[stk[top]],a[i]))<=0)--top;
        stk[++top]=i;
    }
    int p=2;
    rep(i,1,top-1){
        while(X(a[stk[i]],a[stk[i+1]],a[stk[p<top?p+1:1]])>X(a[stk[i]],a[stk[i+1]],a
[stk[p]]))p= p<top?p+1:1;
        ans=max(ans,max(dist(a[stk[i]],a[stk[p]]),dist(a[stk[i+1]],a[stk[p]])));
    }
    printf("%d",ans);
    //printf("\n 内存消耗：%.3f M",(double)sizeof(a)/(1<<20));
    return 0;
}
```

SP4580 ABCDEF - ABCDEF（折半搜索+Hash 表）

```
struct Hash{
    int key,nxt,tot;
}hs[1000055];
int n,a[N],ans,hhs[mod+55],cnt;
void Gethash(int x){
    x+=901000000LL;
    int h=x%mod;
    for(int i=hhs[h];i;i=hs[i].nxt){
        int key=hs[i].key;
        if(key==x){
            ++hs[i].tot;
            return;
        }
    }
    hs[++cnt].key=x;
    hs[cnt].nxt=hhs[h];
```

```
    hs[cnt].tot=1LL;
    hhs[h]=cnt;
}

int queryhash(int x){
    x+=901000000LL;
    int h=x%mod;
    for(int i=hhs[h];i;i=hs[i].nxt){
        int key=hs[i].key;
        if(key==x)return hs[i].tot;
    }
    return 0LL;
}

#undef int
int main()
{
#define int long long
    //freopen("ABCDEF4.in","r",stdin);
    rd(n);
    rep(i,1,n)rd(a[i]);
    rep(i,1,n){
       rep(j,1,n){
          rep(k,1,n){
             Gethash(a[i]*a[j]+a[k]);
           }
        }
    }
    rep(i,1,n){
        if(a[i]==0)continue;
        rep(j,1,n){
            rep(k,1,n){
                ans+=queryhash(a[i]*(a[j]+a[k]));
            }
        }
    }
    printf("%lld\n",ans);
    //printf("内存消耗：%.3f M",(double)sizeof(a)/(1<<20));
    return 0;
}
```

字符串双 Hash 过 KMP

```
#define ull unsigned long long
const ull H1=1e9+7;
const ull H2=1e9+9;
//#define int long long
int n,m,ans=0;
ull sum1[N],sum2[N],h1n=1,h2n=1;
char A[N],B[N];
```

```
//#undef int
int main()
{
//#define int long long
    //freopen("read.txt","r",stdin);
    scanf("%s%s",A+1,B+1);
    n=strlen(A+1),m=strlen(B+1);
    //计算主串滚动 Hash 值,类似前缀和
    rep(i,1,n){
        sum1[i]=sum1[i-1]*H1+(ull)(A[i]-'a');//双 Hash
        sum2[i]=sum2[i-1]*H2+(ull)(A[i]-'a');
    }
    //计算匹配串的 hash
    ull s1=0,s2=0;
    rep(i,1,m){
        s1=s1*H1+(ull)(B[i]-'a');
        s2=s2*H2+(ull)(B[i]-'a');
    }
    //计算 b^n,即 mod^n
    int k=m;
    ull dx=H1;
    while(k){
        if(k&1)h1n=h1n*dx;
        dx*=dx;
        k>>=(ull)1;
    }
    k=m;dx=H2;
    while(k){
        if(k&1)h2n=h2n*dx;
        dx*=dx;
        k>>=(ull)1;
    }
    //匹配
    rep(i,1,n-m+1){
        if(s1==sum1[i+m-1]-sum1[i-1]*h1n&&s2==sum2[i+m-1]-sum2[i-1]*h2n)++ans;
    }
    printf("%d\n",ans);
    //printf("内存消耗：%.3f M",(double)sizeof(sum1)/(1<<20));
    return 0;
}
```

P1950 长方形_NOI 导刊 2009 提高（2）（单调栈模板）

```
#define int long long
char s[N];
int n,m,h[N],stk[N],l[N],r[N],ans=0;

void calc(int i){
    rep(i,1,m){
```

```
        if(s[i]=='.')++h[i];//实际上是 h[i]=h[i-1]+1
        else h[i]=0;
    }
    int top=0;
    dwn(i,m,1){
        while(top&&h[i]<=h[stk[top]]){
            l[stk[top]]=i;--top;
        }
        stk[++top]=i;
    }
    while(top){
        l[stk[top]]=0;
        --top;
    }
    top=0;
    rep(i,1,m){
        while(top&&h[i]<h[stk[top]]){
            r[stk[top]]=i;
            --top;
        }
        stk[++top]=i;
    }
    while(top){
        r[stk[top]]=m+1;
        --top;
    }
    rep(i,1,m){
        ans+=(i-(l[i]+1)+1)*((r[i]-1)-i+1)*h[i];
    }
}
#undef int

int main()
{
#define int long long
    //freopen("read.txt","r",stdin);
    scanf("%d%d",&n,&m);
    rep(i,1,n){
       scanf("%s",s+1);
       calc(i);
    }
    printf("%lld\n",ans);
    return 0;
}
```