动态规划 4

四边形不等式优化

定义

区间包含单调性□$\forall l_1\le l_2\le r_1\le r_2\to f(l_2,r_1)\le f(l_1,r_2)$□

四边形不等式□$\forall l_1\le l_2\le r_1\le r_2\to f(l_1,r_1)+f(l_2,r_2) \le f(l_2,r_1)+f(l_1,r_2)$□

类型一

$$ f_{l,r}=\min_{k=l}^{r-1}(f_{l,k}+f_{k+1,r})+w(l,r) $$

性质一

若 $w(l,r)$ 满足区间包含单调性和四边形不等式，则 $f(l,r)$ 满足四边形不等式。

性质二

记 $g(l,r)$ 为最小最优决策点，即 $f_{l,g(l,r)}+f_{g(l,r)+1,r}=\min_{k=l}^{r-1}(f_{l,k}+f_{k+1,r})$

若 $f(l,r)$ 满足四边形不等式，则 $g(l,r-1)\le g(l,r)\le g(l+1,r)$□

于是状态转移时顺便维护 $g(l,r)$□总时间复杂度 $\sum_{l=1}^n\sum_{r=l+1}^n g(l+1,r)-g(l,r-1)=\sum_{i=1}^n g(i,n)-g(1,i)\le n^2$□

例题

洛谷p1880

题意

给定一个环，环上有 n 堆石头，每次可以合并两堆相邻的石头，费用为两堆石头的数量和，求将所有石头合并到一堆的最小和最大费用。

题解

首先把环倍增成两倍长的链。最小费用状态转移同类型一，易知 w 满足区间包含单调性和四边形不等式。

最大费用考虑贪心，每次都是操作上一次合并的石头堆和与其相邻的石头堆，有 $f_{l,r}=\max(f_{l,r-1}+f_{l+1,r})+w(l,r)$□

```
const int MAXN=205,Inf=1e8;
int dp1[MAXN][MAXN],dp2[MAXN][MAXN],g[MAXN][MAXN],s[MAXN],a[MAXN];
int main()
{
    int n=read_int();
    _rep(i,1,n)a[i]=a[i+n]=read_int();
    _rep(i,1,n*2){
        s[i]=s[i-1]+a[i];
        g[i][i]=i;
    }
    _rep(i,1,2*n)_rep(j,i+1,2*n)dp1[i][j]=Inf;
    for(int i=n*2-1;i;i--)_rep(j,i+1,n*2){
        _rep(k,g[i][j-1],g[i+1][j]){
            if(dp1[i][k]+dp1[k+1][j]+s[j]-s[i-1]<dp1[i][j]){
                dp1[i][j]=dp1[i][k]+dp1[k+1][j]+s[j]-s[i-1];
                g[i][j]=k;
            }
        }
        dp2[i][j]=max(dp2[i+1][j],dp2[i][j-1])+s[j]-s[i-1];
    }
    int ans1=Inf,ans2=0;
    _rep(i,1,n){
        ans1=min(ans1,dp1[i][i+n-1]);
        ans2=max(ans2,dp2[i][i+n-1]);
    }
    enter(ans1);
    enter(ans2);
    return 0;
}
```

类型二

$$ f_{i,j}=\min_{k=1}^{j}(f_{i-1,k}+w(k,r)) $$

性质

若 $w(l,r)$ 满足四边形不等式，记 $g(i,j)$ 为最小最优决策点，则 $g(i,j)\le g(i,j+1)$

考虑 n 次分治法，第 i 次分治法计算 $f_{i,1\sim m}$分治过程维护最小最优决策点的上下界，时间复杂度 $O(nm\log m)$

```
int dp[MAXN][MAXM];
void solve(int pos,int nl,int nr,int ml,int mr){
    if(nl>nr)return;
    int nmid=nl+nr>>1,mmid;
    _rep(i,ml,min(mid,mr)){
        if(dp[pos][nmid]>dp[pos-1][i]+w(i,nmid)){
```

```
                dp[pos][nmid]=dp[pos-1][i]+w(i,nmid);
                mmid=i;
            }
        }
        solve(pos,nl,nmid-1,ml,mmid);
        solve(pos,nmid+1,nr,mmid,mr);
}
```

通式

$$ f_r=\min_{l=1}^{r-1}w(l,r) $$

该式为类型二样例的更加一般化形式，上述解法仍然适用。

类型三

$$ f_r=\min_{l=1}^{r-1}(f_{l}+w(l,r)) $$

性质

若 $w(l,r)$ 满足四边形不等式，则 f 具有决策单调性。记 $g(i)$ 为最小最优决策点，则 $g(i)\le g(i+1)$□

考虑单调队列二分，维护单调队列中每个元素的决策位置 p 和负责的最优决策区间 $[l,r]$□

每次新加入一个点 i□如果该点对序列末尾 n 的决策不如队列末尾的点，则无视该点。

否则和队列末尾的点比较在 l_tail 位置的决策，如果 i 更优则删去末尾的点，不断操作直到 i 不再更优。

最后 i 和队列末尾点的最优决策分界点一定位于区间 $[l_\text{tail},r_\text{tail}]$□二分查找即可。时间复杂度 $O(n\log n)$□

例题

洛谷p3195

题意

给定序列 c 和常数 L□已知一个区间 $[l,r]$ 的权值为 $(\sum_{i=l}^r c_i+r-l-1-L)^2$□现要求将 $[1,n]$ 划分为若干连续区间，使得权值和最小。

题解

设 dp_i 表示区间 $[1,i]$ 的最小答案，设 $s_n=\sum_{i=1}^n a_n$□可以得到状态转移方程

$$ \text{dp}_i=\min(dp_j+(s_i+i-s_j-j-L-1)^2) $$

设 $w(l,r)=(s_r+r-s_l-l-L-1)^2$，不难发现 $w(l,r)$ 满足四边形不等式，直接套用即可。

```
const int MAXN=5e4+5;
LL s[MAXN],dp[MAXN],m;
struct Seg{
    int lef,rig,idx;
    Seg(int lef=0,int rig=0,int idx=0):lef(lef),rig(rig),idx(idx){}
}que[MAXN];
LL w(int l,int r){return (s[r]+r-s[l]-l-1-m)*(s[r]+r-s[l]-l-1-m);}
LL cal(int l,int r){return dp[l]+w(l,r);}
int cutSeg(int lef,int rig,int idx1,int idx2){
    int ans;
    while(lef<=rig){
        int mid=lef+rig>>1;
        if(cal(idx1,mid)<cal(idx2,mid)){
            ans=mid;
            lef=mid+1;
        }
        else
        rig=mid-1;
    }
    return ans;
}
int main()
{
    int n=read_int(),head=1,tail=0;
    m=read_int();
    _rep(i,1,n)s[i]=read_int()+s[i-1];
    que[++tail]=Seg(1,n,0);
    _rep(i,1,n){
        dp[i]=cal(que[head].idx,i);
        while(head<=tail&&que[head].rig<=i)head++;
        que[head].lef=i+1;
        if(i<n&&cal(i,n)<=cal(que[tail].idx,n)){
while(head<=tail&&cal(que[tail].idx,que[tail].lef)>=cal(i,que[tail].lef))ta
il--;
            if(head<=tail){
                int p=cutSeg(que[tail].lef,que[tail].rig,que[tail].idx,i);
                que[tail].rig=p;
                que[++tail]=Seg(p+1,n,i);
            }
            else
            que[++tail]=Seg(i+1,n,i);
        }
    }
    enter(dp[n]);
    return 0;
}
```

补充性质

1. 若 w_1,w_2 均满足四边形不等式(或区间包含单调性)□$c_1,c_2\ge 0$□则 $c_1w_1+c_2w_2$ 满足四边形不等式(或区间包含单调性)
2. 若存在 $f(x),g(x)$ 使得 $w(l,r)=f(r)-g(l)$□则函数 $w(l,r)$ 满足四边形恒等式。若 $f(x),g(x)$ 单增，则 $w(l,r)$ 满足区间包含单调性。
3. 设 $h(x)$ 是一个凸函数，若函数 $w(l,r)$ 满足四边形恒等式并且对区间包含关系具有单调性，则复合函数 $h(w(l,r))$ 也满足四边形不等式。