

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;
}
```

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92_4&rev=1619601479

Last update: 2021/04/28 17:17