动态规划 2

二进制优化

题意

洛谷p1776

给定一个容量为 m 的背包和 n 种物品。每种物品价值为 v_i，重量为 w_i，数量为 c_i，求最多可以得到的价值。

题解

暴力方法为将每种物品转化为 c_i 个独立物品考虑，然后就是一个 01 背包问题。

不难发现任意一个不超过 c_i 的正整数一定用 $1,2,4\cdots 2^n,(c_i-2^{n+1}+1)$ 表示。

于是可以将每种物品拆成 $1,2,4\cdots 2^n,(c_i-2^{n+1}+1)$ 个物品构成的包考虑。

物品总数优化为 $O\left(\sum_{i=1}^n \log {c_i}\right)$，总时间复杂度为 $O\left(m\sum_{i=1}^n \log {c_i}\right)$。

```
const int MAXM=4e4+5,MAXC=2005;
int dp[MAXM],v[MAXC],w[MAXC];
int main()
{
    int n=read_int(),m=read_int(),cnt=0;
    _for(i,0,n){
        int a=read_int(),b=read_int(),c=read_int();
        for(int j=1;j<c;j<<=1){
            v[cnt]=j*a,w[cnt++]=j*b;
            c-=j;
        }
        v[cnt]=c*a,w[cnt++]=c*b;
    }
    _for(i,0,cnt)
    for(int j=m;j>=w[i];j--)
    dp[j]=max(dp[j],dp[j-w[i]]+v[i]);
    enter(dp[m]);
    return 0;
}
```

单调队列优化

习题一

CF372C

题意

一个城镇有 n 个区域，从左到右 1 编号为 n 每个区域之间距离 1 个单位距离。

节日中有 m 个烟火要放，给定放的地点 a_i 时间 t_i 如果你当时在区域 x 那么你可以获得 $b_i-\vert a_i-x\vert$ 的开心值。

你每个单位时间可以移动不超过 d 个单位距离。你的初始位置是任意的(初始时刻为 1)，求你通过移动能获取到的最大的开心值。

题意

设 $\text{dp}(i,j)$ 表示在 t_i 时刻处于位置 j 能得到的最大快乐值。于是有

$$ \text{dp}(i,j)=\max_{j-d(t_i-t_{i-1})\le k\le j+d(t_i-t_{i-1})}\text{dp}(i-1,k)+b_i-\vert a_i-j\vert $$

单调队列维护即可。时间复杂度 $O(nm)$

```
const int MAXN=15e4+5;
const LL Inf=1e18;
LL dp[2][MAXN];
int que[MAXN];
int main()
{
    int n=read_int(),m=read_int(),d=read_int(),pos=0,last=1;
    while(m--){
        int a=read_int(),b=read_int(),t=read_int()-last;
        pos=!pos;
        for(int i=1,rpos=1,head=0,tail=1;i<=n;i++){
            while(head>=tail&&que[tail]<i-1LL*t*d)tail++;
            while(rpos<=n&&rpos<=i+1LL*t*d){
                while(head>=tail&&dp[!pos][que[head]]<=dp[!pos][rpos])head-
-;
                que[++head]=rpos++;
            }
            dp[pos][i]=dp[!pos][que[tail]]+b-abs(a-i);
        }
        last+=t;
    }
    LL ans=-Inf;
    _rep(i,1,n)
    ans=max(ans,dp[pos][i]);
    enter(ans);
```

```
    return 0;
}
```

习题二

题意

洛谷p1776

给定一个容量为 m 的背包和 n 种物品。每种物品价值为 v_i，重量为 w_i，数量为 c_i，求最多可以得到的价值。

题解

考虑滚动背包，有 $\text{dp}_i=\max_{k\le c}(\text{dp}_{i-kw}+kv)$，设 $i=jw+r$，有

$$ \text{dp}_{jw+r}=\max_{k\le c}(\text{dp}_{jw+r-kw}+kv)=\max_{k\le c}(\text{dp}_{(j-k)w+r}-(j-k)v)+jv $$

枚举余数 r，对每个余数 r，维护 $\text{dp}_{aw+r}-av$ 的单调队列即可。时间复杂度 $O(nm)$。

```
const int MAXM=4e4+5,MAXC=2005;
int dp[MAXM];
pair<int,int> que[MAXM];
int main()
{
    int n=read_int(),m=read_int();
    _for(i,0,n){
        int v=read_int(),w=read_int(),c=read_int();
        _for(j,0,w){
            if(m<j)continue;
            int pos1=(m-j)/w,pos2=pos1,head=0,tail=1;
            for(int k=0;pos2>=0&&k<=c;pos2--,k++){
                int t=dp[pos2*w+j]-pos2*v;
                while(head>=tail&&que[head].second<=t)head--;
                que[++head]=make_pair(pos2,t);
            }
            for(;pos1>=0;pos1--){
                while(que[tail].first>pos1)tail++;
                dp[pos1*w+j]=max(dp[pos1*w+j],que[tail].second+pos1*v);
                if(pos2<0)continue;
                int t=dp[pos2*w+j]-pos2*v;
                while(head>=tail&&que[head].second<=t)head--;
                que[++head]=make_pair(pos2,t);
                pos2--;
            }
        }
    }
```

```
    enter(dp[m]);
    return 0;
}
```

习题三

洛谷p2254

题意

给定一个 $n\times m$ 的图，图中有若干障碍物。接下来给定一个物体的初始位置和 k 个时间段。

每个时间段内物体将按每个单位时间一格的速率向给定方向运动。

每个单位时间都可以选择让物体在该单位时间停止运动。问在物体移动过程中不超出图的边界同时不撞上障碍物的前提下物体可以运动的最大路程。

题解

设 $\text{dp}(i,x,y)$ 表示物体在前 i 个时间段最终运动到 (x,y) 能运动的最大路程。以向 x 增大的方向运动为例，有状态转移方程

$$ \text{dp}(i,x,y)=\max_{x-dt\le j\le x}(\text{dp}(i-1,j,y)+x-j)=\max_{x-dt\le j\le x}(\text{dp}(i-1,j,y)-j)+x $$

其中 dt 表示该时间段的长度。发现可以单调队列维护，总时间复杂度 $O(nmk)$

```
const int MAXN=205;
int n,m,dp[MAXN][MAXN],pos;
pair<int,int> que[MAXN];
char mp[MAXN][MAXN];
void cal(int x,int y,int dx,int dy,int dt){
    for(int i=1,head=0,tail=1;1<=x&&x<=n&&1<=y&&y<=m;i++,x+=dx,y+=dy){
        if(mp[x][y]=='x'){
            head=0,tail=1;
            continue;
        }
        while(head>=tail&&que[tail].first<i-dt)tail++;
        int cur=dp[x][y]-i;
        while(head>=tail&&que[head].second<=cur)head--;
        que[++head]=make_pair(i,cur);
        dp[x][y]=que[tail].second+i;
    }
}
int main()
{
    n=read_int(),m=read_int();
```

```
    int x=read_int(),y=read_int(),t=read_int();
    _rep(i,1,n)
    scanf("%s",mp[i]+1);
    mem(dp,0xf0);
    dp[x][y]=0;
    while(t--){
        int l=read_int(),r=read_int(),dr=read_int();
        if(dr==1)_rep(i,1,m)cal(n,i,-1,0,r-l+1);
        else if(dr==2)_rep(i,1,m)cal(1,i,1,0,r-l+1);
        else if(dr==3)_rep(i,1,n)cal(i,m,0,-1,r-l+1);
        else _rep(i,1,n)cal(i,1,0,1,r-l+1);
    }
    int ans=0;
    _rep(i,1,n)_rep(j,1,m)ans=max(ans,dp[i][j]);
    enter(ans);
    return 0;
}
```

斜率优化

习题一

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

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

考虑将变量 i,j 分离，设 $a_i=s_i+i,b_i=s_i+i+L+1$，有

$$ \text{dp}_i=\text{dp}_j+(a_i-b_j)^2=a_i^2+\text{dp}_j+b_j^2+2a_ib_j $$

设 $y=\text{dp}_j+b_j^2,x=2b_j$，于是有

$$ \text{dp}_i-a_i^2=y=a_ix $$

于是维护凸包即可，另外发现 a_i 和 x 递增，于是可以单调队列维护凸包，时间复杂度 $O(n)$，注意最开始需加入 (a_0,b_0)。

```
const int MAXN=5e4+5;
LL dp[MAXN],a[MAXN],b[MAXN],s[MAXN];
int q[MAXN];
double slope(int i,int j){return (double)(dp[i]+b[i]*b[i]-dp[j]-
b[j]*b[j])/(2*b[i]-2*b[j]);}
int main()
{
    int n=read_int(),L=read_int();
    b[0]=L+1;
    _rep(i,1,n){
        s[i]=s[i-1]+read_int();
        a[i]=s[i]+i;
        b[i]=a[i]+L+1;
    }
    int head=0,tail=-1;
    q[++tail]=0;
    _rep(i,1,n){
        while(head<tail&&slope(q[head],q[head+1])<a[i])head++;
        dp[i]=dp[q[head]]+(a[i]-b[q[head]])*(a[i]-b[q[head]]);
        while(head<tail&&slope(q[tail],i)<slope(q[tail-1],q[tail]))tail--;
        q[++tail]=i;
    }
    enter(dp[n]);
    return 0;
}
```