遥远的金字塔（pyramid）

80 分做法：

本题阶段性明显,可以每一层为一阶段,用 DP 解决。

设 F[i][j]表示已经选择了 j 个矩形,最后一个顶部在第 i 层的最大覆盖面积，w[i]为第 i 层宽度。

则有状态转移方程:

$F[i][j]=\max\{F[k][j-1]+w[i]*(i-k)\}(1\leqslant k\leqslant i-1)$

初始化 F[i][1]=w[i]*i,最后在 F[K]~F[N]中取一个最大值即可,时间复杂度为 $O(N^2K)$.

100 分做法：

考虑优化。因为 w[i]从底层往上走是不增的,故考虑斜率优化。

由于方程只与前一个状态有关,则可以设 F[i]表示已经选择了 p 个矩阵,G[i]表示已选择 p-1 个矩形,i 同上,则状态转移方程可改写为:

$F[i]=\max\{G[j]+w[i]*(i-j)\}$设有决策点 k,j(k<j),若决策 j 优于 k,则有:

$G[j]+w[i]\times(i-j)>G[k]+w[i]\times(i-k)$

$\Rightarrow G[j]+w[i]\times i-w[i]\times j > G[k]+w[i]\times i-w[i]\times k$

$\Rightarrow G[j]-G[k] > (j-k)\times w[i]$

$\because k < j \therefore$ 设 $S(i,j)=\dfrac{G[j]-G[k]}{j-k}$

则有 S(i,j)>w[i]。

结论 1:对于 i 的两个决策点 k,j(k<j),如果有 S(j,k)>w[i],则决策 j 比 k 优,反之,k 比 j 优。

结论 2:对于三个决策点 k,j,i(k<j<i),如果有 S(j,k)<S(i,j),则决策 j 永远不会成为最优决策。

那么,我们可以用一个单调队列来维护决策,利用以上决策剔除无用决策点。

①对于队首,如果 S(q[L+1],q[L])>w[i],说明 q[L+1]比 q[L]优,去掉 q[L],一直到不满足条件为止。

②此时 q[L]就是当前点的最优决策,根据状态转移方程计算 F[i]的值。

③对于队尾,如果 S(q[R],q[R-1])<S(i,q[R]),证明 q[R]永远不会成为最优决策,去掉 q[R],一直到不满足条件为止,然后将决策 i 入队。

最后要提醒的是,在计算斜率时,为了避免精度带来的误差,要将除法变成乘法;结果可能超过 int 范围,要用 64 位整形。