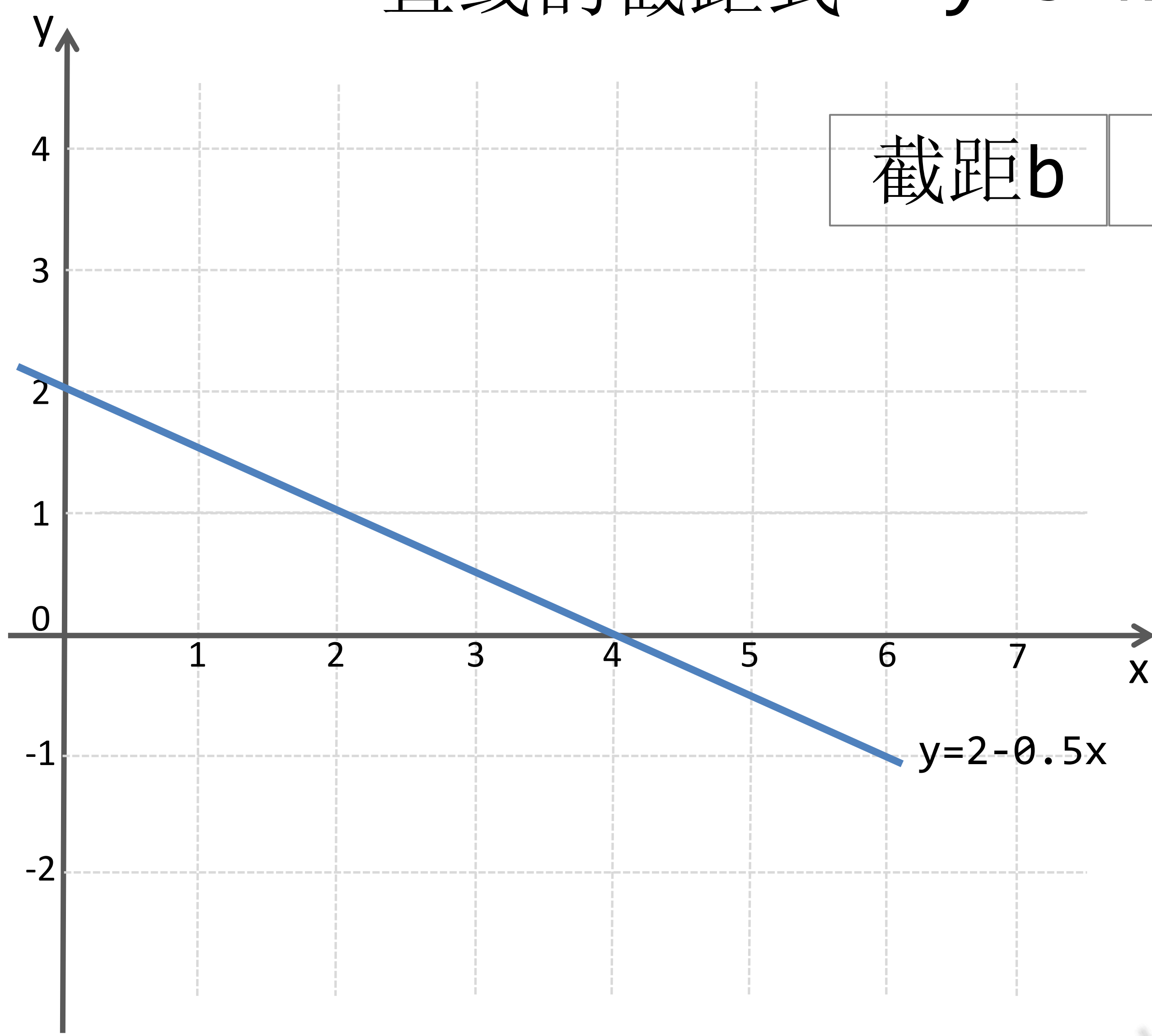
C++算法

直线表达式

直线的截距式　$y=b-kx$



截距b	斜率k

缺点：截距式无法表示竖直线

请求出交点坐标

1号直线 $y=b_1+k_1x$

2号直线 $y=b_2+k_2x$

若$k_1=k_2$且$b_1=b_2$，两直线重合

若$k_1=k_2$且$b_1\neq b_2$，两直线平行

若$k_1\neq k_2$，两直线有唯一交点

两式相减

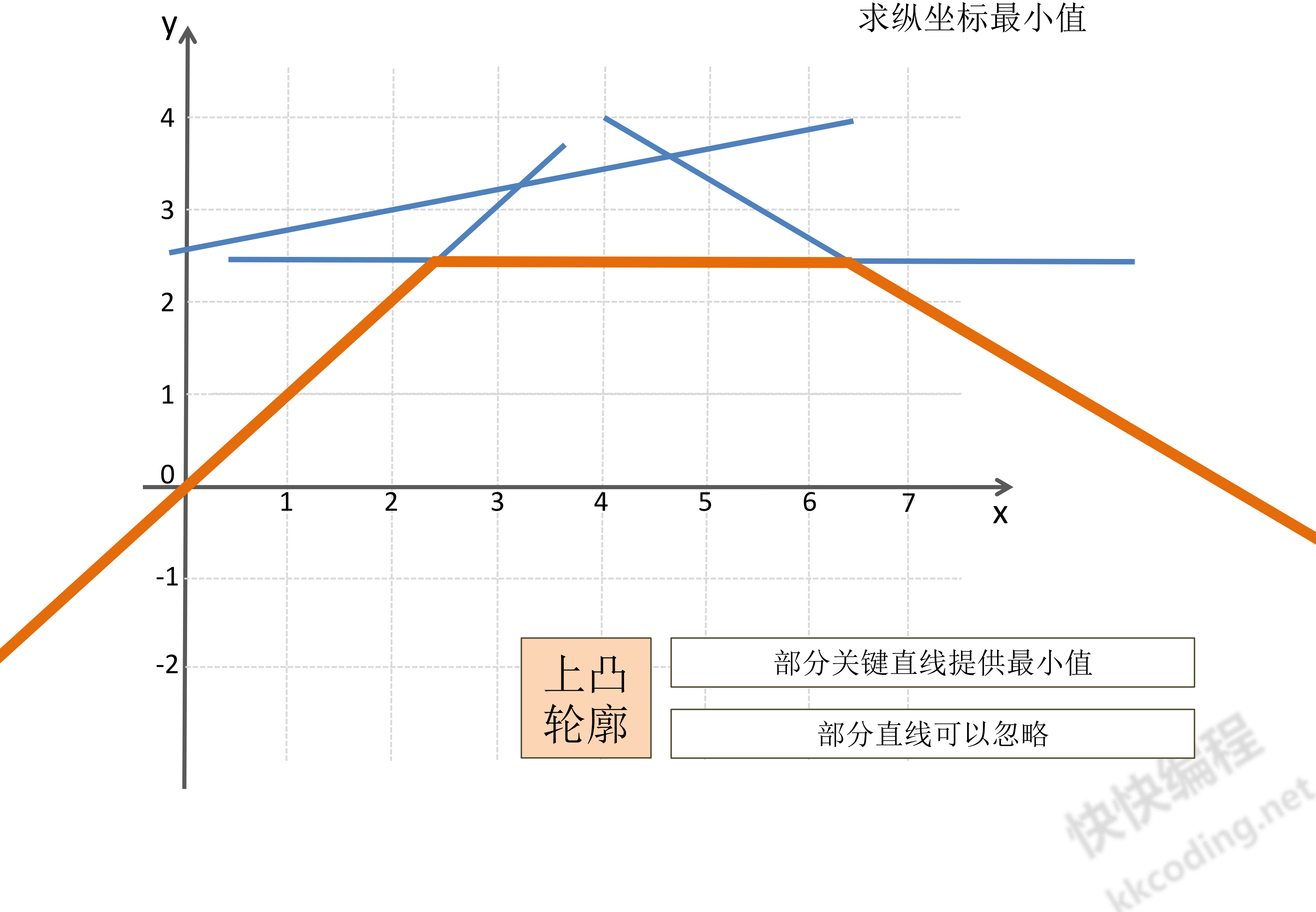
$$0=(b_1-b_2)+(k_1-k_2)x$$

$$x=-(b_1-b_2)/(k_1-k_2)$$

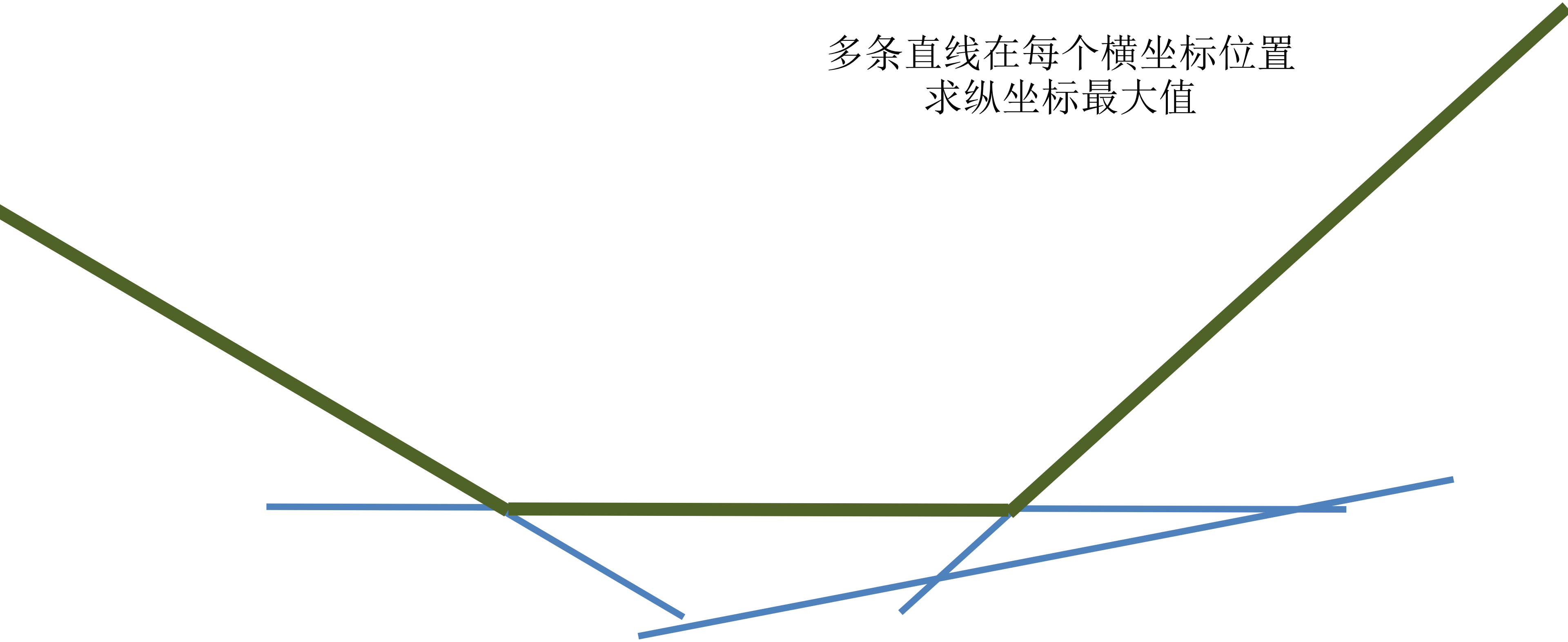
$$y=(k_1b_2-k_2b_1)/(k_1-k_2)$$

直线最值

多条直线在每个横坐标位置
求纵坐标最小值



多条直线在每个横坐标位置
求纵坐标最大值



下凸
轮廓

部分关键直线提供最大值

部分直线可以忽略

快快编程2627

暴力

```
struct Line{ll b,k;} lines[N];
```

```
    ll n,m;
    cin>>n>>m;
    for(ll i=1;i<=n;++i)cin>>lines[i].b;
    for(ll i=1;i<=n;++i)cin>>lines[i].k;
    for(ll i=1;i<=m;++i)cin>>x[i];
```

```
    for(ll i=1;i<=m;++i){
        ans[i]=INF;
        for(ll j=1;j<=n;++j)
            ans[i]=min(ans[i],lines[j].b+lines[j].k*x[i]);
    }
```

所有直线最小值
形成上凸函数

$$y = \min_{i=1,2,\ldots,n} \{ b_i + k_i x \}$$

第一步

哪些直线参与形成最小值

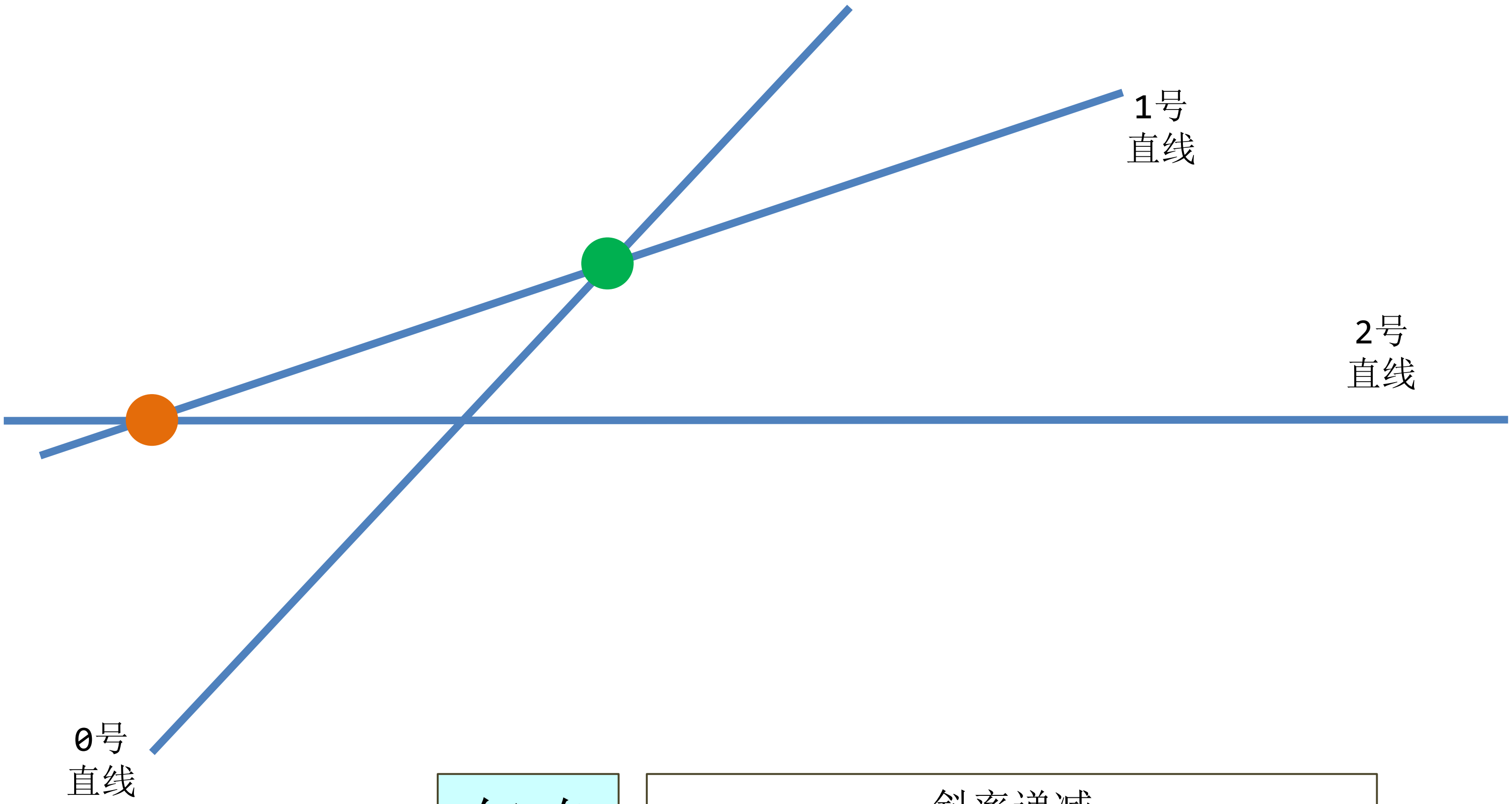
哪些直线可以忽略

第二步

从小到大考虑各个横坐标
对应的纵坐标最小值

第一步

哪些直线参与形成最小值
哪些直线可以忽略



忽略无效直线

斜率递减
2号和1号直线的交点在 1号和0号直线的交点左侧
1号直线不会参与最小值,可以忽略

```cpp
struct Line{ll b,k;} lines[N];
ld X(ll u,ll v){
    return -(ld)(lines[u].b-lines[v].b)/(lines[u].k-lines[v].k);
}
```

队列q

维护可能参与最小值的直线编号

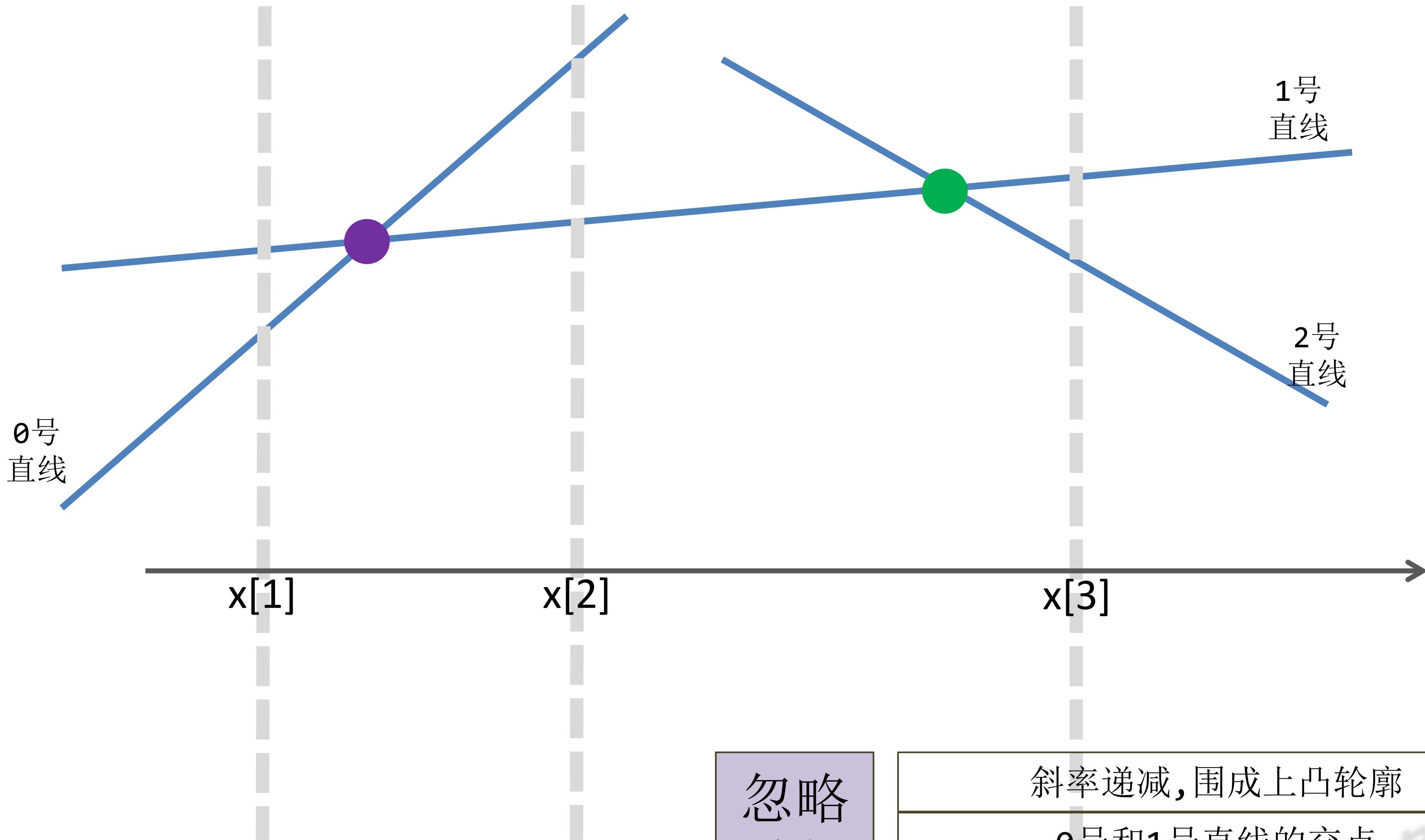
```cpp
    ll l=1,r=1;
    for(ll i=1;i<=n;++i){
        while(r-l>=2&&X(i,q[r-1])<X(q[r-1],q[r-2]))--r;
        q[r++]=i;
    }
```

r-l表示队列里直线数量

队列里原本至少有2条直线时
i号直线才可能删除某一条

第二步

从小到大考虑各个横坐标对应的纵坐标最小值



忽略过气直线

斜率递减,围成上凸轮廓
0号和1号直线的交点在x[2]左侧
0号直线过气,不再提供最值答案

```
struct Line{ll b,k;} lines[N];
ld X(ll u,ll v){
	return -(ld)(lines[u].b-lines[v].b)/(lines[u].k-lines[v].k);
}
```

```
	ll l=1,r=1;
	for(ll i=1;i<=n;++i){

		q[r++]=i;
	}
```

```
	for(ll i=1;i<=m;++i){

		int j=q[l];
		ans[i]=lines[j].b+lines[j].k*x[i];
	}
```

最优决策

模型	序列分段/切割问题
状态	f[i]表示前i本书最小的占据面积
决策	上一段的结尾编号j
转移方程	

$$f[i] = \min_{j=0\ldots i-1} \{ f[j] + w[j+1] \times h[i] \}$$

朴素方法	枚举所有可能的j

```cpp
struct Book{ll h,w;} bk[N];

    cin>>n;
    for(ll i=1;i<=n;++i)
        cin>>bk[i].h>>bk[i].w;

    for(ll i=1;i<=n;++i){
        f[i]=bk[1].w*bk[i].h;
        for(ll j=1;j<i;++j)
            f[i]=min(f[i],f[j]+bk[j+1].w*bk[i].h);
    }
```

复杂度$O(n^2)$	如何加速

DP的凸优化

DP的斜率优化

凸优化

模型	序列分段/切割问题
状态	f[i]表示前i本书最小的占据面积
决策	上一段的结尾编号j

$$f[i] = \min_{j=0\ldots i-1} \{ f[j] + w[j+1] \times h[i] \}$$

转移方程	转换为 直线取最值问题
j号直线	y=f[j]+w[j+1]*x

截距	f[j]	
斜率	w[j+1]	递减
计算f[i]时取点 只依赖i不依赖j	横坐标h[i]	递增

凸优化

模型	序列分段/切割问题
状态	f[i]表示前i本书最小的占据面积
决策	上一段的结尾编号j

$$f[i] = \min_{j=0\dots i-1} \{ f[j] + w[j+1] \times h[i] \}$$

转移方程	转换为 直线取最值问题
j号直线	y=f[j]+w[j+1]*x
f[i]含义	对0到i-1号直线 在横坐标h[i]处取最小值
队列q	维护可能参与最小值的直线编号

```
16 struct Book{ll h,w;} bk[N];
17 struct Line{ld b,k;} lines[N];
18 ld X(ll u,ll v){
19     return -(ld)(lines[u].b-lines[v].b)/(lines[u].k-lines[v].k);
20 }
```

```
29     lines[0]=(Line){________};
30     q[1]=0;
31     ll l=1,r=2;
```

0号直线截距和斜率

0号直线入队

队内直线个数为r-1

```
32     for(ll i=1;i<=n;++i){
33         while(______________________)++l;
34         ll j=q[l];
35         f[i]=f[j]+bk[j+1].w*bk[i].h;
36         lines[i]=(Line){f[i],bk[i+1].w};
37         while(______________________)--r;
38         ________
39     }
```

左删

最优决策

i号直线截距和斜率

右删

i号直线入队

第36,37行能否交换

不能

快快编程829

模型	序列分段/切割问题
状态	f[i]表示只能用前i个选址 覆盖前i个景点的最小代价
决策	上一段的结尾编号j

转移方程

$$f[i] = \min_{j=0\ldots i-1} \{ f[j] + \frac{(i-j)(i-j-1)}{2} \} + a[i]$$

$$= \min_{j=0\ldots i-1} \{ f[j] + \frac{(i-1)i + (j+1)j - 2ij}{2} \} + a[i]$$

$$= \min_{j=0\ldots i-1} \{ f[j] + \frac{(j+1)j}{2} - ij \} + a[i] + \frac{(i-1)i}{2}$$

模型	序列分段/切割问题
状态	f[i]表示只能用前i个选址覆盖前i个景点的最小代价

$$f[i]=\min_{j=0\ldots i-1}\left\{f[j]+\frac{(j+1)j}{2}-ij\right\}+a[i]+\frac{(i-1)i}{2}$$

转移方程	转换为 直线取最值问题
j号直线	$y=f[j]+\frac{(j+1)j}{2}-jx$

截距	$f[j]+\frac{(j+1)j}{2}$	
斜率	$-j$	递减
计算f[i]时取点 只依赖i不依赖j	横坐标i	递增

```
1 /*姓名XXX
2 f[i]表示只能用前i 个选址覆盖前i 个景点的最小代价
3      i= 1, 2, 3, 4, 5, 6, 7, 8, 9,10
4 a[i]= 2  3  1  5  4  5  6  3  1  2
5 f[i]= 2  4  4  9  9 12 16 15 16 18
6 */
7 /*
8 f[i]=min{f[j]+(i-j)(i-j-1)/2}+a[i]
9       =min{f[j]+(i*i-i*j-i-i*j+j*j+j)/2}+a[i]
10      =min{f[j]+j*(j+1)/2-i*j}+a[i]+i*(i-1)/2
11
12 j 号直线: y=f[j]+j*(j+1)/2-j*i
13 截距b=f[j]+j*(j+1)/2, 不一定单调
14 斜率k=-j, 递减
15 计算f[i]时取点横坐标i, 递增
16 */
```

请同学完成第7-16行

快快编程

	2627,2628,829
拓展题	814,815