

太戈编程  
etiger.vip

# 信奥算法

# 太戈编程2668

```
30 /*
31  $f[i]$  表示打包前  $i$  件物品的最小费用
32  $n=5, G=4$ 
33  $i= 1, 2, 3, 4, 5$ 
34  $w[i]= 3 \quad 4 \quad 2 \quad 1 \quad 4$ 
35  $f[i]=$ 




```

$f[i]$ 表示打包前 $i$ 件物品的最小费用

序列切割的典型决策

最后一段有多长

前一段结尾在哪里

转移方程

$$f[i] = \min_{j=0,1,\dots,i-1} \{f[j] + cost(j+1, i)\}$$

## 朴素DP

```
22 void DP(){
23     for(ll i=1;i<=n;++i){
24         f[i]=INF;
25
26
27     }
28     cout<<f[n]<<endl;
29 }
```

如何加速

凸优化  
斜率优化

将 $f[i]$ 的转移方程理解为  
多条直线在特定点取最小值的问题

这些直线围成的轮廓为上凸函数

$$f[i] = \min_{j=0,1,\dots,i-1} \{ \text{截距}(j) + \text{斜率}(j) \times \text{横坐标}(i) \} + \text{其他}(i)$$

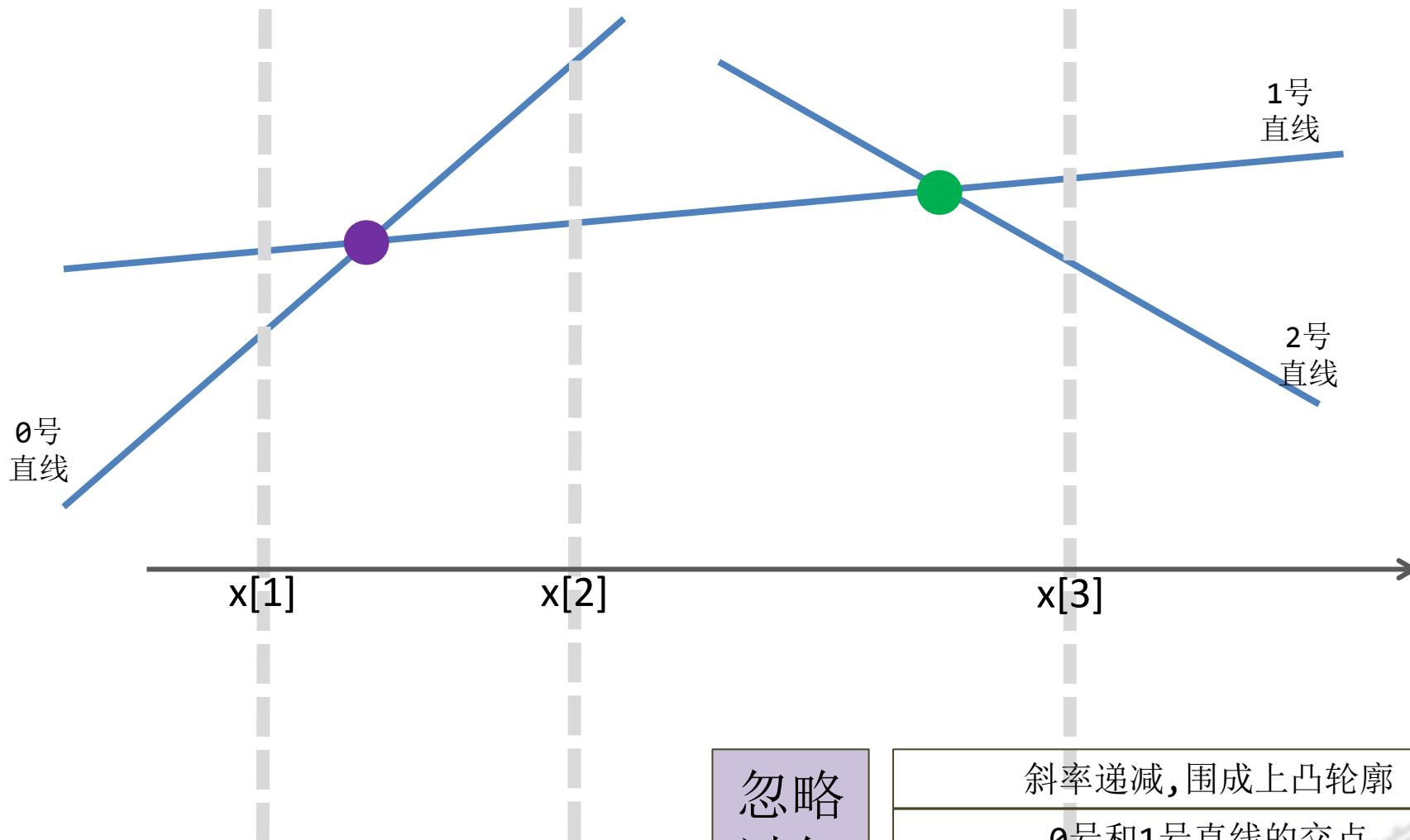
单调  
递减



单调  
递增



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

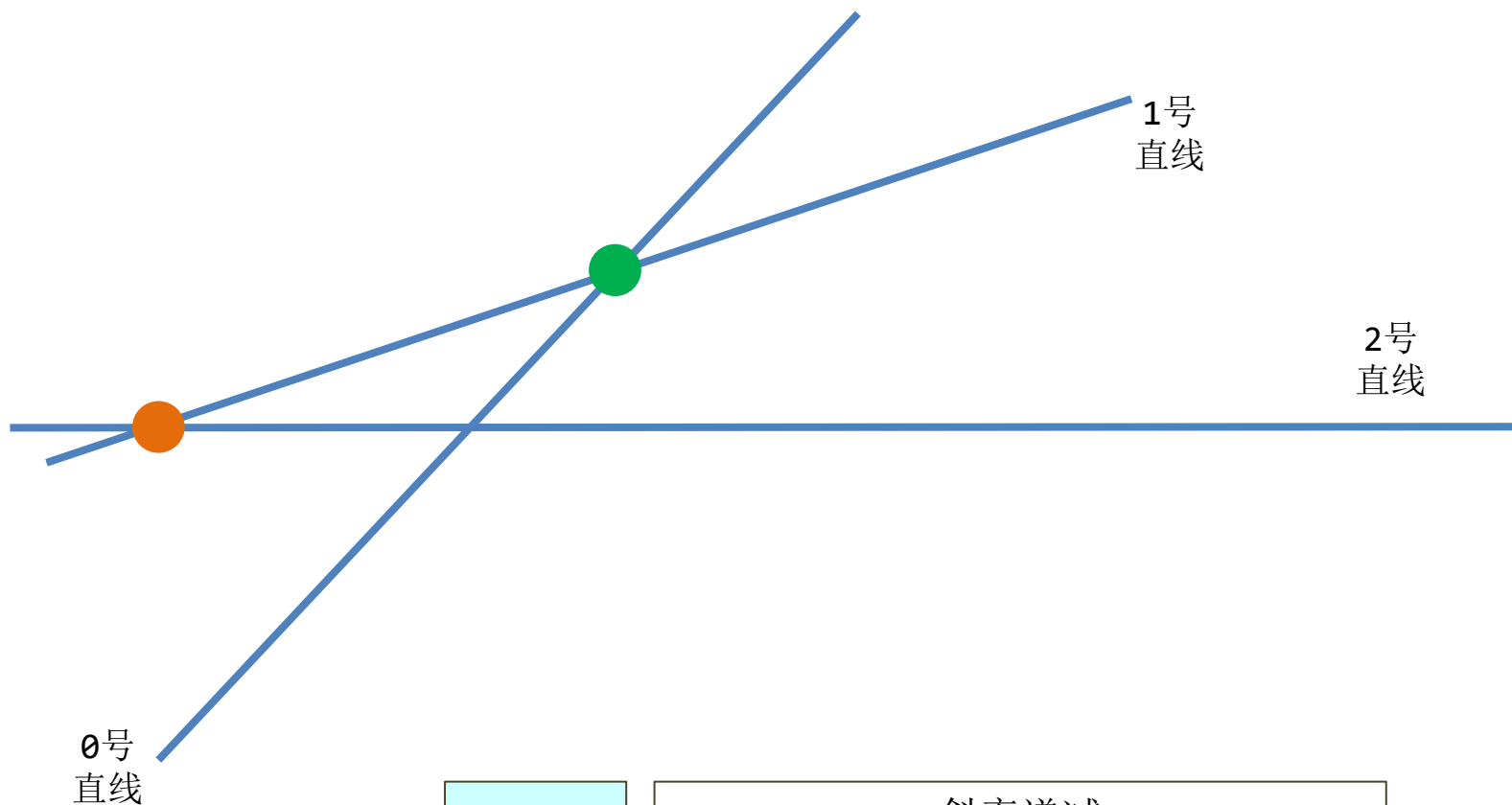
0号和1号直线的交点  
在 $x[2]$ 左侧

0号直线过气, 不再提供最值答案



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

```
37  $f[i] = \min\{f[j] + \text{cost}(j+1, i)\}$ 
38  $f[i] = \min\{f[j] + (s[i] - s[j] + i - j - 1 - G)^2\}$ 
39     令  $t[i] = s[i] + i$ 
40     令  $H = G + 1$ 
41  $f[i] = \min\{f[j] + (t[i] - t[j] - H)^2\}$ 
42  $f[i] = \min\{f[j] + t[j]^2 + 2 * H * t[j] - 2 * t[j] * t[i]\}$ 
43      $+ t[i]^2 + H * H - 2 * t[i] * H$ 
44
45  $j$  号直线:  $y = f[j] + t[j]^2 + 2 * H * t[j] - 2 * t[j] * x$ 
46 计算  $f[i]$  时取点横坐标  $t[i]$ , 递增
47
48  $i$  号直线:  $y = f[i] + t[i]^2 + 2 * H * t[i] - 2 * t[i] * x$ 
49 截距  $b =$   递增
50 斜率  $k =$   递减
51  $*/$ 
```

```
47 struct Line{ld b,k;} lines[N];  
48 ld X(ld u,ld v){  
49     return  
50     |  
51 }
```

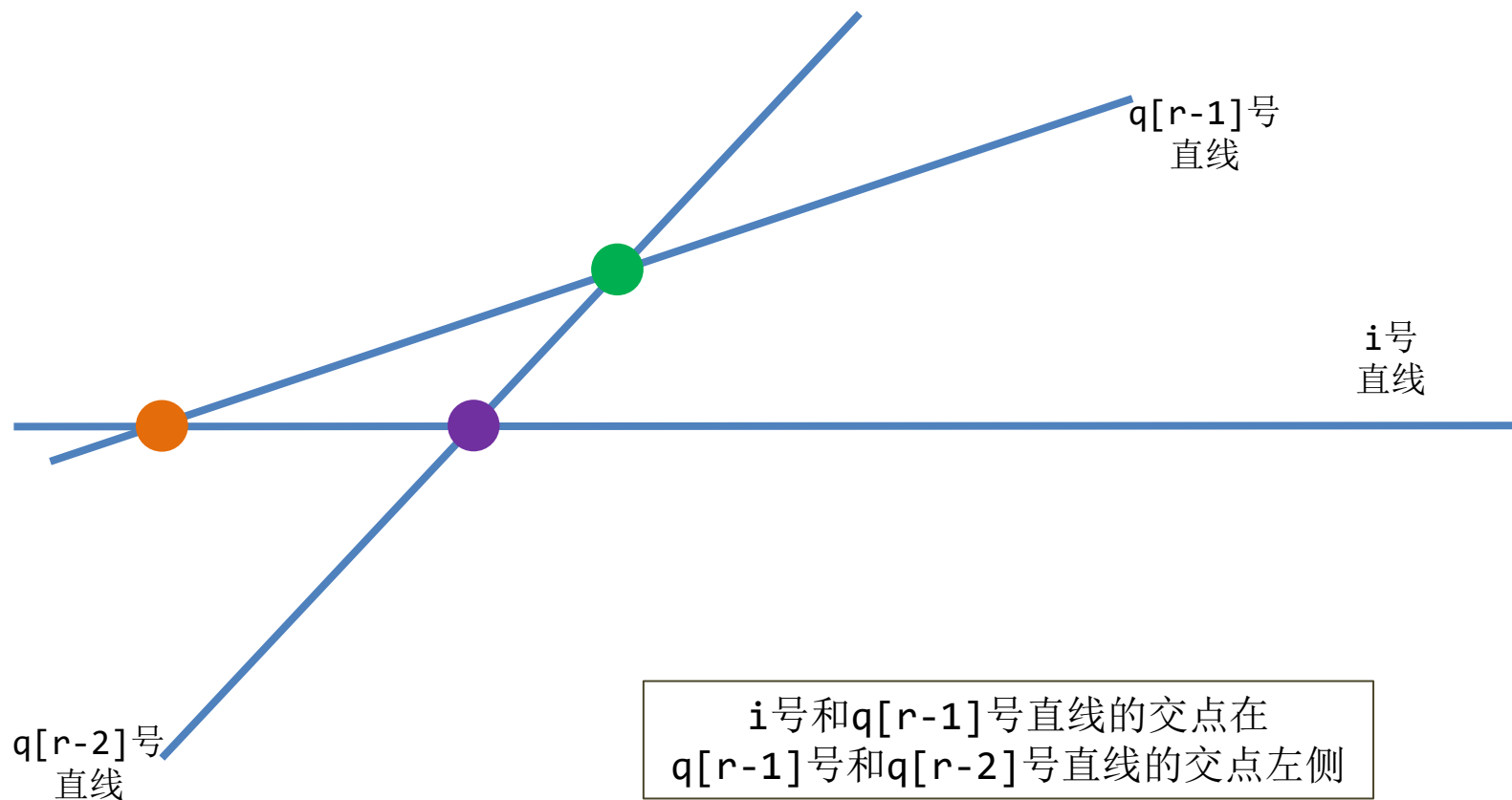
```

52 void DPconvex(){
53     lines[0]=(Line) 
54     q[1]=0;
55     
56     for(ll i=1;i<=n;++i){
57         while( )++l;
58         ll j=q[l];
59         f[i]=
60         lines[i]=(Line) 
61         while( )--r;
62         q[r++]=i;
63     }
64     cout<<f[n]<<endl;
65 }
    
```

右删有多种写法

哪些直线参与形成最小值

哪些直线可以忽略



i号和q[r-1]号直线的交点在  
q[r-1]号和q[r-2]号直线的交点左侧

i号和q[r-2]号直线的交点在  
q[r-1]号和q[r-2]号直线的交点左侧

i号和q[r-1]号直线的交点在  
i号和q[r-2]号直线的交点左侧

## 讨论题

截距是否单调对算法有哪些影响?

没有影响

# 太戈编程2669

```
30  /*
31   $f[i]$  表示只装修前i层楼满足前i层需求的最小费用
32   $n=4$ 
33       $i=$     1,    2,    3,    4,
34   $c[i]=$     2     4     2     4
35   $x[i]=$     3     1     4     2
36   $f[i]=$ 




```



$f[i]$ 表示只装修前*i*层楼  
满足前*i*层需求的最小费用

序列切割的典型决策

最后一段有多长

前一段结尾在哪里

转移方程

$$f[i] = \min_{j=0,1,\dots,i-1} \{f[j] + cost(j+1,i)\} + c[i]$$

```
30  /*
31   $f[i]$  表示只装修前i层楼满足前i层需求的最小费用
32   $n=4$ 
33       $i=$   1,   2,   3,   4,
34   $c[i]=$   2   4   2   4
35   $x[i]=$   3   1   4   2
36   $f[i]=$   2   6   5   9
37
38   $f[i]=\min\{f[j]+cost(j+1,i)\}+c[i]$ 
39   $f[i]=\min\{f[j]+x[i-1]*1+x[i-2]*2+...+x[j+1]*(i-j-1)\}+c[i]$ 
40      令  $t[i]=t[i+1]+x[i]$ 
41      令  $g[i]=g[i+1]+x[i]*(n-i+1)$ 
42   $f[i]=$  
```

如何加速

凸优化  
斜率优化

```
34  $f[i] = \min\{f[j] + x[i-1]*1 + x[i-2]*2 + \dots + x[j+1]*(i-j-1)\} + c[i]$ 
35      $\hookrightarrow t[i] = t[i+1] + x[i]$ 
36      $\hookrightarrow g[i] = g[i+1] + x[i]*(n-i+1)$ 
37  $f[i] = \min\{f[j] + g[j+1] - g[i] - (t[j+1] - t[i])*(n-i+1)\} + c[i]$ 
38  $f[i] = \min\{f[j] + g[j+1] - t[j+1]*(n-i+1)\}$ 
39      $+ c[i] - g[i] + t[i]*(n-i+1)$ 
40  $f[i] = \min\{f[j] + g[j+1] - t[j+1]*(n+1) + t[j+1]*i\}$ 
41      $+ c[i] - g[i] + t[i]*(n-i+1)$ 
42
43
44
45
46
47
48
49 */
```

纯i相关项基本用不到

```

55 void DPconvex(){
56     lines[0]=(Line)·[ ];
57     q[1]=0;
58     ll l=1,r=2;
59     for(ll i=1;i<=n;++i){
60         while([ ])++l;
61         ll j=q[l];
62         f[i]=[ ];
63         lines[i]=(Line)·[ ];
64         while([ ])--r;
65         q[r++]=i;
66     }
67     printf("%lld\n",f[n]);
68 }

```

哪几行可以改变顺序?

## 2669另解

$$\text{cost}(j + 1, i) = \sum_{p=j+1}^{i-1} x[p] \times (i - p)$$

$$= i \times \sum_{p=j+1}^{i-1} x[p] - \sum_{p=j+1}^{i-1} x[p] \times p$$

$$= i \times (s[i - 1] - s[j]) - (h[i - 1] - h[j])$$

$$s[i] = \sum_{p=1}^i x[p]$$

$$h[i] = \sum_{p=1}^i x[p] \times p$$

```

25  /*
26  f[i]表示只装修前i层楼满足前i层需求的最小费用
27  n=4
28      i=   1,   2,   3,   4,
29  c[i]=   2   4   2   4
30  x[i]=   3   1   4   2
31  f[i]=   2   6   5   9
32
33  f[i]=min{f[j]+cost(j+1,i)}+c[i]
34  f[i]=min{f[j]+ sum_{p=j+1..i-1} {x[p]*(i-p)}+c[i]
35  f[i]=min{f[j]+ sum_{p=j+1..i-1} {x[p]}*i-sum_{p=j+1..i-1} {x[p]*p}}+c[i]
36      令s[i]=s[i-1]+x[i]
37      令h[i]=h[i-1]+x[i]*i
38  f[i]=min{f[j]+(s[i-1]-s[j])*i-(h[i-1]-h[j])}+c[i]
39  f[i]=min{f[j]+h[j]-s[j]*i}+c[i]+s[i-1]*i-h[i-1]
40
41  j号直线:y=f[j]+h[j]-s[j]*x
42  计算f[i]时取点横坐标i, 递增
43
44  i号直线方程:y=f[i]+h[i]-s[i]*x
45  i号直线截距b=f[i]+h[i],
46  i号直线斜率k=-s[i], 递减
47  */

```

## 讨论题

若问题要求最大值  
算法如何修改



## 查错方法

打印DP数组f[ ]  
打印直线数组lines[ ]

## 现场挑战2670

请完成朴素DP

请完成DP凸优化/斜率优化分析

请完成对拍



# 太戈编程

2668, 2669, 2670