

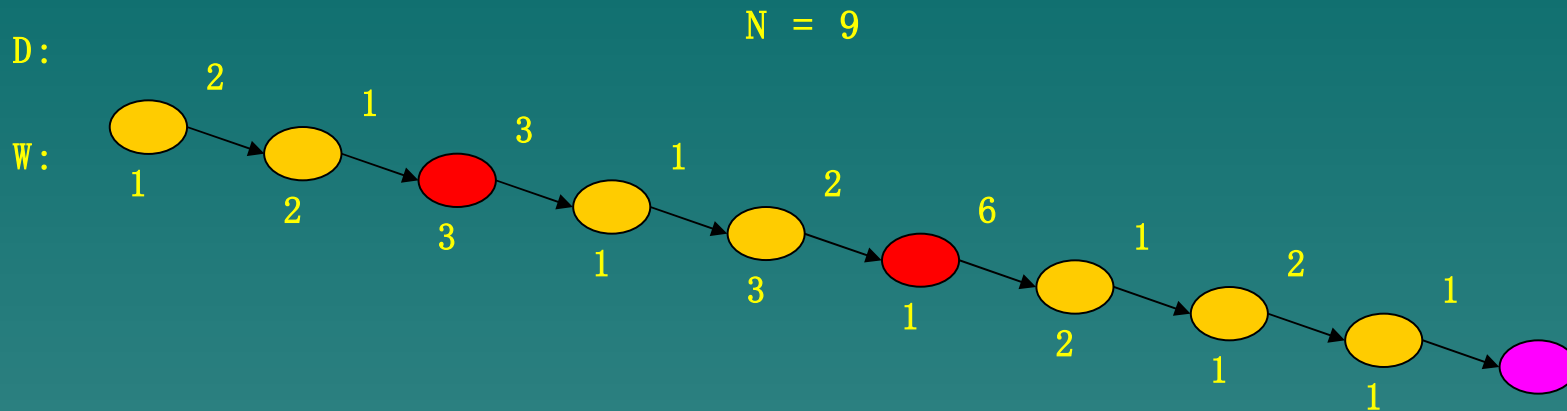
# 从一类单调性问题看 算法的优化

长沙市一中 汤泽

充分挖掘数据关系，灵活运用数据结构，往往  
是构造出优秀算法的关键因素

- 一般队列：一端插入，另一端删除
- 特殊队列：尾端插入，两端删除
- 单调性：帮助优化一类单调性问题

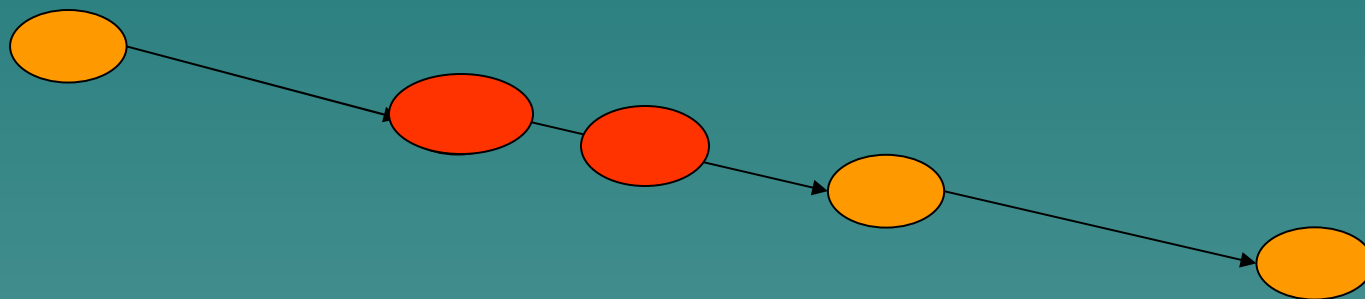
# 问题 1 锯木厂选址



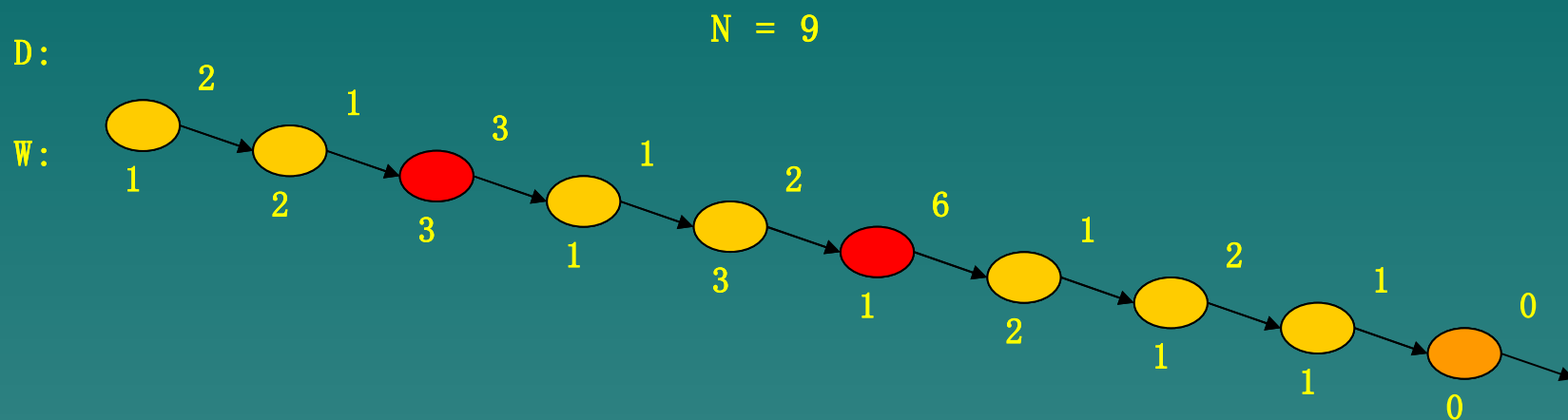
$2 \leq N \leq 20000, 1 \leq W_i \leq 10000, 1 \leq D_i \leq 10000$

# 问题 1 锯木厂选址

最优方案中，锯木厂必定建在有树的位置



# 问题 1 锯木厂选址



	1	2	3	4	5	6	7	8	9	10
Sd	0	2	3	6	7	9	15	16	18	19
Sw	1	3	6	7	10	11	13	14	15	15

# 问题 1 锯木厂选址

分析：

- **C[I]:** 在第 I 棵树处建立一个锯木厂, 并且将第 1 到第 I 棵树全部运到这个锯木厂所需的费用  
$$C[I] = C[I-1] + Sw[I-1] * D[I-1], \quad C[0] = 0$$
- **W[J,I]:** 在第 I 棵树处建立一个锯木厂, 并且将第 J+1 到第 I 棵树全部运往这个锯木厂的费用  
$$W[J,I] = C[I] - C[J] - Sw[J] * (Sd[I] - Sd[J])$$
  
当时 J ,  $W[J,I] = 0$

# 问题 1 锯木厂选址

分析：

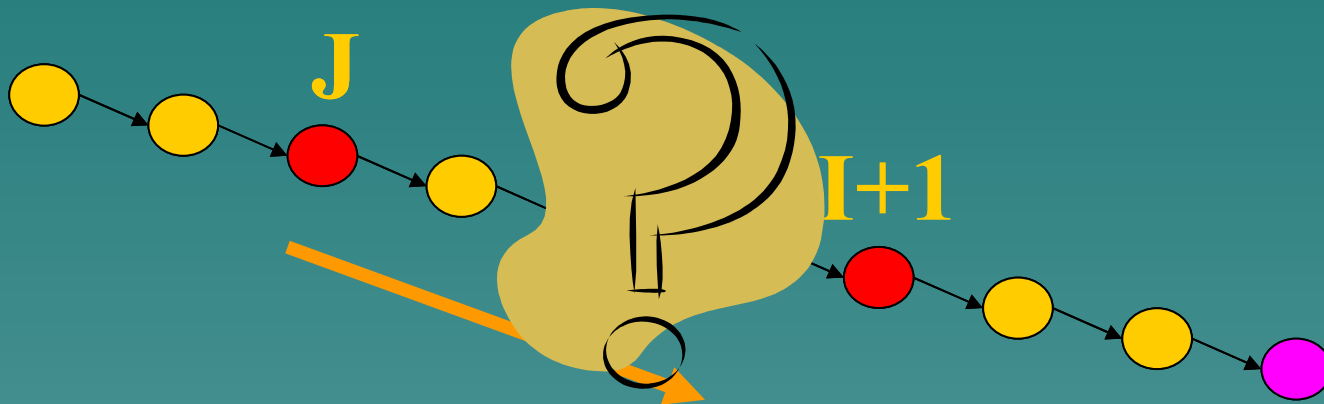
$F[I]$  表示在第  $I$  棵树处建立第二个锯木厂的最小费用，则：

$$f[i] = \min_{1 \leq j \leq i-1} \{c[j] + w[j, i] + w[i, n+1]\}$$

这个算法的时间复杂度为  $O(N^2)$

# 问题 1 锯木厂选址

$$f[i] = \min_{1 \leq j \leq i-1} \{c[j] + w[j, i] + w[i, n+1]\}$$





# 问题 1 锯木厂选址

证明：

➤ 令  $S[K,I]$  表示决策变量取  $K$  时  $F[I]$  的值

$$S[K,I] = C[K] + W[K,I] + W[I,N+1]$$

$$W[J,I] = C[I] - C[J] - Sw[J] * (Sd[I] - Sd[J])$$

➤ 设  $K1 < K2 < I$ ,  $S[K1,I] - S[K2,I] < 0$

$$(Sw[K1] * Sd[K1] - Sw[K2] * Sd[K2]) /$$

$$(Sw[K1] - Sw[K2]) > Sd[I]$$

# 问题 1 锯木厂选址

证明：

- 设  $K1 < K2 < I$ ,  $S[K1, I] - S[K2, I] < 0$

$$(Sw[K1] * Sd[K1] - Sw[K2] * Sd[K2]) /$$

$$(Sw[K1] - Sw[K2]) > Sd[I]$$

- 令  $g[K1, K2] =$  上面不等式的左边

$$S[k1, I] - S[k2, I] < 0 \hat{=} g[K1, K2] > Sd[I]$$

- 因为  $D[K] \geq 0$ , 因此  $Sd$  序列不下降

$$Sd[I'] > g[K1, K2] > Sd[I] \quad (I < I')$$

# 问题 1 锯木厂选址

分析：

维护一个特殊队列  $K, K_1 < K_2 < \dots < K_n$ ,  
 $g[K_1, K_2] < g[K_2, K_3] < \dots < g[K_{n-1}, K_n]$ :

- 计算状态  $F[I]$  前, 若  
 $g[K_1, K_2] \leq Sd[I]$ , 表示决策  $K_1$  不比  
 $K_2$  优, 删除  $K_1$ , 重复该步骤

# 问题 1 锯木厂选址

分析：

- 计算  $F[I]$ ,  $F[I] = C[K_1] + W[K_1, I] + W[I, n+1]$
- 若  $g[K_{n-1}, K_n] > g[K_n, I]$ ,  
 $Sd[I'] > g[K_{n-1}, K_n] > g[K_n, I]$   
 $K_n$  比  $K_{n-1}$  优之前  $I$  就将比  $K_n$  优，删除  $K_n$ , 重复该步骤，最后将  $I$  插入队列

算法总复杂度  $O(N)$

## 问题 2 旅行问题

问题描述：

一个环形跑道上有  $n$  个加油站，按顺时针编号

为 1 到  $n$  ( $3 < n < 10^6$ )

第  $i$  号加油站有  $P_i$  ( $0 \leq P_i < 10^9$ ) 升汽油，

每升汽油可供行驶一千米

## 问题 2 旅行问题

一个例子：

**N=5**

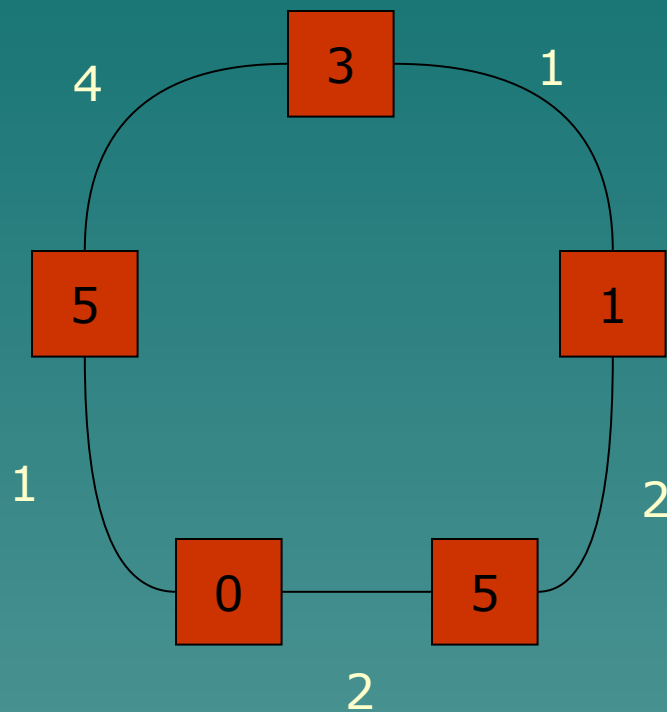
**P[1]=3; D[1]=1**

**P[2]=1; D[2]=2**

**P[3]=5; D[3]=2**

**P[4]=0; D[4]=1**

**P[5]=5; D[5]=4**



## 问题 2 旅行问题

一个例子：

**N=5**

**P[1]=3; D[1]=1**

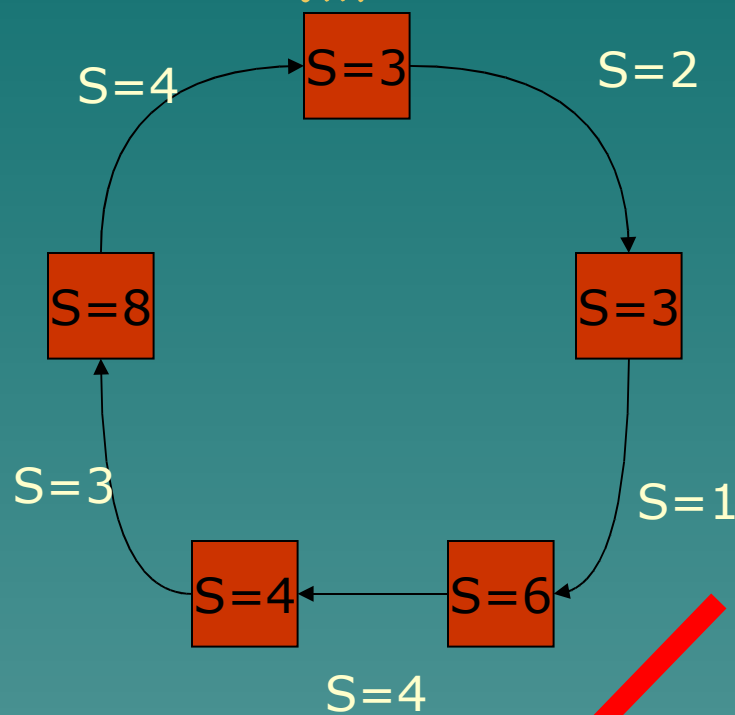
**P[2]=1; D[2]=2**

**P[3]=5; D[3]=2**

**P[4]=0; D[4]=1**

**P[5]=5; D[5]=4**

以 1 号加油站为起  
点



## 问题 2 旅行问题

以 2 号加油站为起点

一个例子：

**N=5**

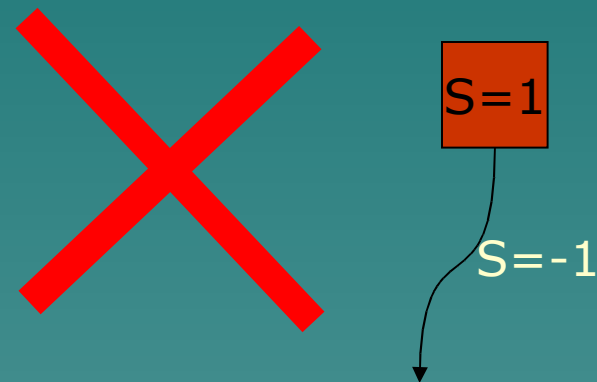
**P[1]=3; D[1]=1**

**P[2]=1; D[2]=2**

**P[3]=5; D[3]=2**

**P[4]=0; D[4]=1**

**P[5]=5; D[5]=4**





## 问题 2 旅行问题

一个例子：

**N=5**

**P[1]=3; D[1]=1**

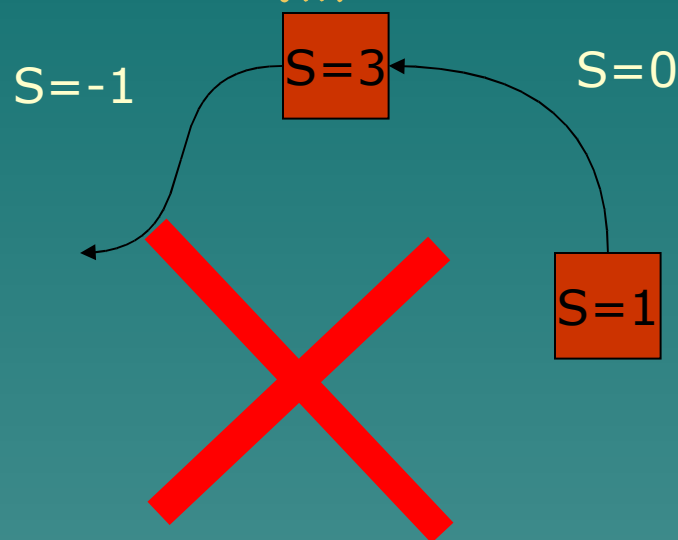
**P[2]=1; D[2]=2**

**P[3]=5; D[3]=2**

**P[4]=0; D[4]=1**

**P[5]=5; D[5]=4**

以 2 号加油站为起  
点



## 问题 2 旅行问题

一个例子：

**N=5**

**P[1]=3; D[1]=1**

**P[2]=1; D[2]=2**

**P[3]=5; D[3]=2**

**P[4]=0; D[4]=1**

**P[5]=5; D[5]=4**

以 **1、3、5** 号加油站为起点有办法周游一圈

# 算法 一

分析：

- 直接模拟刚刚的演算过程

太慢了！

- 总的时间复杂度为  $O(N^2)$

但是  $N$  最大可以达到  $10^6$ !

# 算法 二

分析：

- 假定只能按顺时针方向行驶。
- 令  $A[I] = P[I] - D[I]$   
 $A[I+N] = A[I]$   
 $A[0] = 0$
- 设  $SA[I]$  表示  $A$  序列中前  $I$  项的和

# 算法 二

	1	2	3	4	5
P	3	1	5	0	5
D	1	2	2	1	4

	0	1	2	3	4	5	6	7	8	9
A	0	2	-1	3	-1	1	2	-1	3	-1
sa	0	2	1	4	3	4	6	5	8	7



# 算法 二

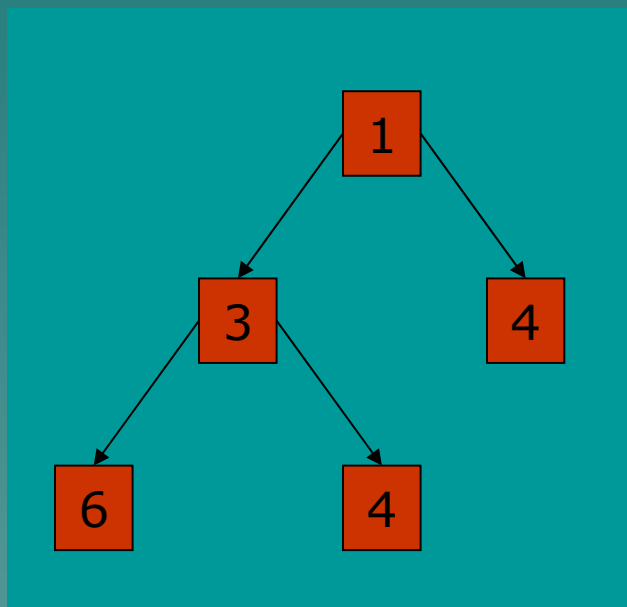
分析：

- $SA[I]$  到  $SA[I+N-1]$  都不小于  $SA[I-1]$
- $SA[I]$  到  $SA[I+N-1]$  中的最小数不小于  $SA[I-1]$

求  $N$  个数中的最小数，很自然地想到了堆！

# 算法 二

	0	1	2	3	4	5	6
A	0	2	-1	3	-1	1	2
Sa	0	2	1	4	3	4	6



- 堆中不超过 **N** 个元素
- **2N** 次插入操作
- **2N** 次删除操作
- **N** 次取堆顶元素
- 总复杂度  **$O(N\log_2 N)$**

# 算法 三

SA: 0 2 1 4 3 4 2 1 4 3



分析：

- 给定一个序列 **SA** 和 **N** 个区间
- 求出每个区间中的最小值，对其作相应判定

**这是一个标准的 RMQ 问题！**

- 时间复杂度降为  **$O(N)$**



# 算法 四

分析：

给定的  $N$  个区间不存在包含关系，满足单调性！

**同样满足单调性？**

0 1 2 3 4 5 6 7 8 9

Sa:

0	2	1	4	3	4	2	1	4	3
---	---	---	---	---	---	---	---	---	---



K

2 2 7 7 7

:

# 算法 四

预处理：

- 维护一个特殊队列  $K$ ,  $K_1 < K_2 < \dots < K_n$   
 $Sa[k_1] < Sa[k_2] < \dots < Sa[K_n]$
- 将  $1, 2, \dots, n-1$  依次插入队列  $K$ . 插入前, 如果  $Sa[i] \leq Sa[k_n]$ , 将  $K_n$  删除. 反复该步骤, 最后将  $i$  插入队列

# 算法 四

求解并更新  $K$ :

- 若  $K_1=i-1$ , 将  $K_1$  删除出队列
- 插入新位置号  $i+n-1$ ,  
若  $Sa[K_n] \geq Sa[i+n-1]$ , 删除  $K_n$ ,  
重复这个步骤, 最后将  $i+n-1$  作为  
 $K_n$  插入队列
- 若  $Sa[k_1] \geq Sa[i-1]$ , 表示从  $i$  号加油站出发可以周游一周
- 总复杂度  $O(N)$

# 四个算法比较

	空间复杂度	时间复杂度	实现难度
算法一	$O(N)$	$O(N^2)$	很容易
算法二	$O(N)$	$O(N \log_2 N)$	简单
算法三	$O(N)$	$O(N)$	难
算法四	$O(N)$	$O(N)$	简单

# 总 结

通过充分挖掘数据关系，发现隐含的单调性，  
以较低的编程复杂度成功地实现了算法的优化

- 注意问题的特殊性
- 学会灵活变通

总 结

善于发现

勇于创新

谢谢大家！

