

# 极客时间算法训练营

## 第二十课

### 实战技巧、总结与回顾

李煜东

《算法竞赛进阶指南》作者



# 目录

1. 位运算
2. 期末串讲



# 位运算

# 为什么需要位运算？

机器采用二进制对数值进行表示、存储和运算

在程序中恰当地使用二进制，可以提高运行效率

十进制与二进制的转化

- 短除法
- 幂次和
- $6_{(10)} = 1*2^2 + 1*2^1 + 0*2^0 = 110_{(2)}$

$$\begin{array}{ccccccc} 6 & = & 4 & + & 2 & + & 0 \\ & & \nearrow & & \uparrow & \uparrow & \\ & & 2^2 & & 2^1 & & 2^0 \end{array}$$

$$6_{(10)} = 110_{(2)}$$

除数	商	余数
2	6	0
2	3	1
2	1	1
	0	

↑

# 位运算符

+ 进位 int x = 1;  
-2 ~x = (1111 ... 10)<sub>2</sub>

含义	运算符	示例
按位与	&	1011 0011 → 0011
按位或		1011 0011 → 1011
按位取反	~	0011 → 1100 (4位二进制数)
按位异或	^	1011 0011 → 1000 (相同得0 不同得1)
左移	<<	0011 → 0110 00110
右移	>>	0011 → 0001 0001x

移位  
(补符号位)

12下取整

0011

0001

$$(-2)_{10} = (1111 \dots 10)_2 = (11 \dots 1)_2 = (-1)_{10}$$

# 异或运算 (xor)

相同为 0, 不同为 1

也可用“不进位加法”来理解

异或运算的特点与应用:

- $x \wedge 0 = x$

- $x \wedge x = 0$

- 结合律:  $a \wedge b \wedge c = a \wedge (b \wedge c) = (a \wedge b) \wedge c$

- 成对变换:  $x \wedge 1$  ( $0 \wedge 1 = 1$ ,  $1 \wedge 1 = 0$ ;  $2 \wedge 1 = 3$ ,  $3 \wedge 1 = 2$ ;  $4 \wedge 1 = 5$ ,  $5 \wedge 1 = 4$ )

- 交换两个数:  $a \leftarrow a \wedge b$ ,  $b \leftarrow a \wedge b$ ,  $a \leftarrow a \wedge b$

`int temp = x; x = y; y = temp;`

$a \wedge b$   $a \wedge b$   $a \wedge b$

$a = a \wedge b$   
 $b = a$

$$x \wedge 1 = x$$

$$\begin{array}{r} 1\ 1\ 0\ 0\ 1 \\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

$$\begin{array}{r} 1\ 1\ 0\ 0\ 1 \\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

# 指定位置的位运算

二进制数的**最右边**为第 0 位

获取 x 在二进制下的第 n 位 (0 或者 1) :  $(x \gg n) \& 1$

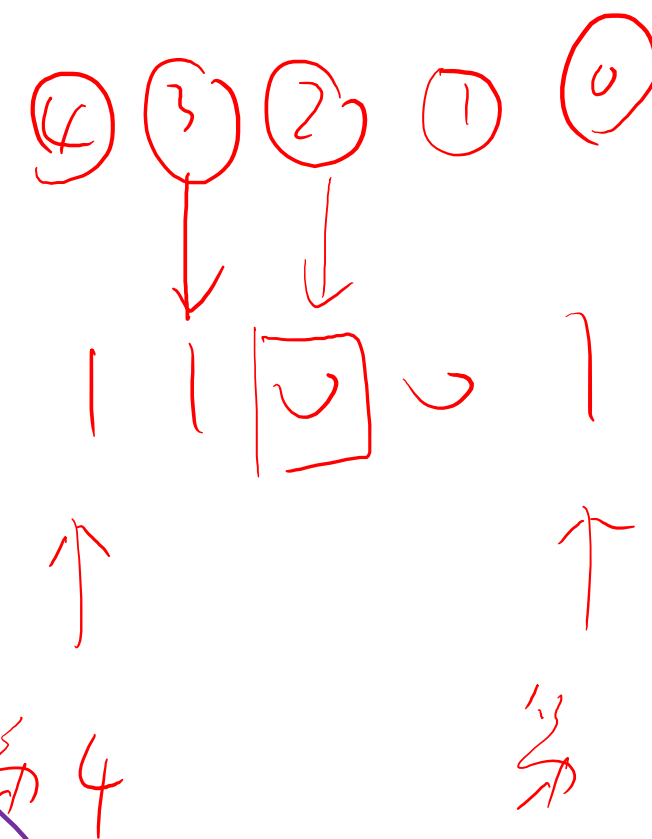
将 x 在二进制下的第 n 位置为 1:  $x | (1 \ll n)$

将 x 在二进制下的第 n 位置为 0:  $x \& (\sim(1 \ll n))$

将 x 在二进制下的第 n 位取反:  $x \wedge (1 \ll n)$

将 x 最右边的 n 位清零:  $x \& (\sim 0 \ll n)$

将 x 最高位至第 n 位 (含) 清零:  $x \& ((1 \ll n) - 1)$



3 2

1 0 0 0 1

1 0 1 1 0 1 1 0 X

1 1 1 1 0 0 0 0

1 0 1 1 0 0 0 0 ✓

1 1 1 1 1 1 1 1

$(00010000)_2 - (1)_2$   
 $00001111$

# 位运算实战要点

判断奇偶：

- $x \% 2 == 1 \rightarrow (x \& 1) == 1$
- $x \% 2 == 0 \rightarrow (x \& 1) == 0$

除以 2 的幂次：

- $x / 2 \rightarrow x \gg 1$
- $\text{mid} = (\text{left} + \text{right}) / 2; \rightarrow \text{mid} = (\text{left} + \text{right}) \gg 1;$

$$\text{left} + (\text{right} - \text{left}) / 2$$

lowbit:

- 得到最低位的 1:  $x \& -x$  或  $x \& (\sim x + 1)$
- 清零最低位的 1:  $x = x \& (x - 1)$

$$\begin{array}{r} 10100 - 1 \\ \hline 10011 \\ \hline 10000 \end{array}$$



# 实战

位 1 的个数

<https://leetcode-cn.com/problems/number-of-1-bits/>

2 的幂

<https://leetcode-cn.com/problems/power-of-two/>

颠倒二进制位

<https://leetcode-cn.com/problems/reverse-bits/>

比特位计数

<https://leetcode-cn.com/problems/counting-bits/>

# 实战：位运算优化

Pow(x, n) —— 快速幂

<https://leetcode-cn.com/problems/powx-n/>

N 皇后

<https://leetcode-cn.com/problems/n-queens/>

解数独

<https://leetcode-cn.com/problems/sudoku-solver/>

# 期末串讲

# 高频主题

计算机科学基础：数据结构、树/图、DFS/BFS

- 第 1、3、5、6、7、14、16 课

状态空间与子问题划分

- 递归、分治、动态规划的设计
- 第 4、6、10、11、12、13 课

决策与候选集合优化

- “滑动窗口”系列的多种维护方法，固定一端+移动一端+消除冗余的思想
- 第 2、13 课



# 高端技巧

Pre-calculation: 空间换时间（前缀和等）

- 第 2 课

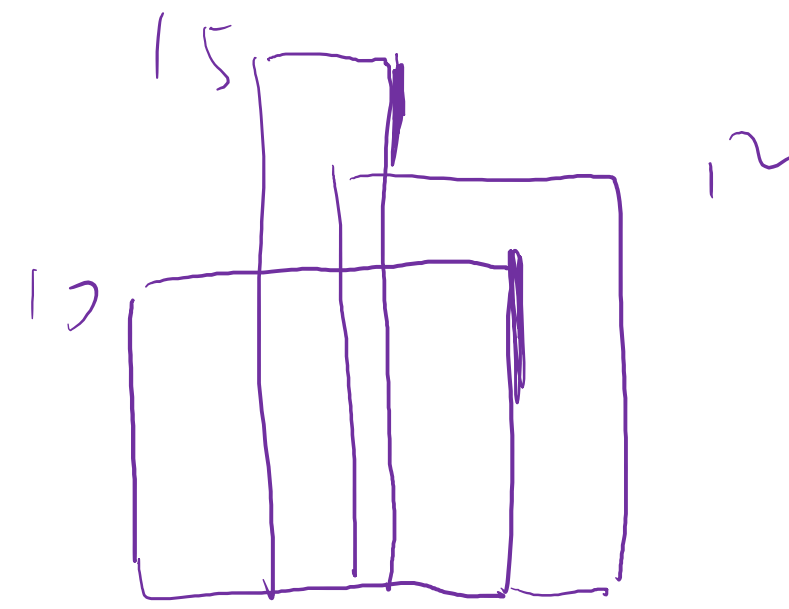
二分答案：求解转化为判定

- 第 8 课

离线批处理，关键事件思想

- 第 9、20 课

# 实战



天际线问题

<https://leetcode-cn.com/problems/the-skyline-problem/>

批处理 + 差分思想

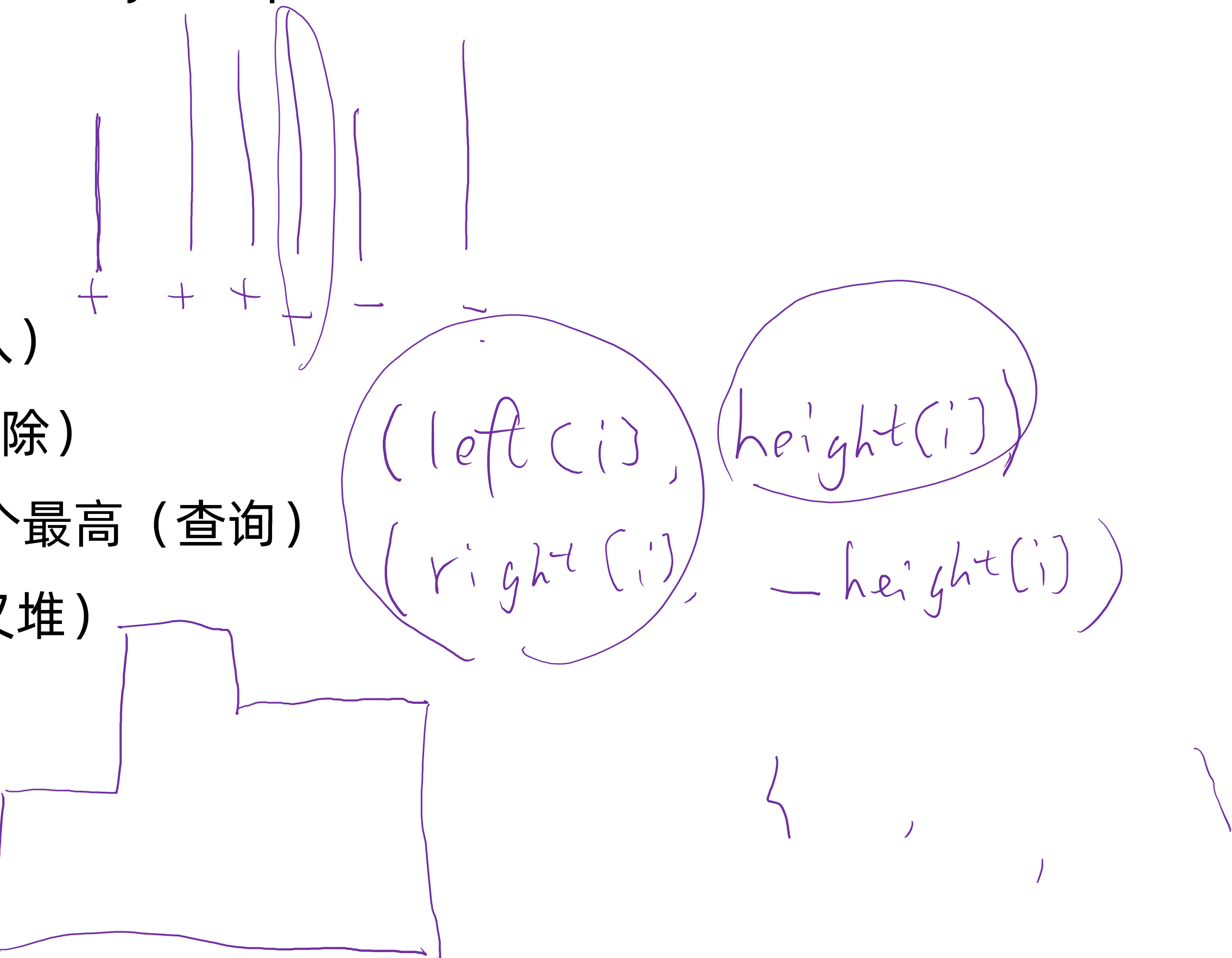
每个建筑物拆成两个事件：

- 在  $\text{left}[i]$  处,  $\text{height}[i]$  开始可用（插入）
- 在  $\text{right}[i]$  处,  $\text{height}[i]$  不再可用（删除）

任何时刻只需要判断可用的  $\text{height}$  里哪个最高（查询）

排序 + 数据结构（set, 或懒惰删除的二叉堆）

vs. 支持区间修改的线段树



# 实战

包含每个查询的最小区间

<https://leetcode-cn.com/problems/minimum-interval-to-include-each-query/>

批处理所有询问，得到答案后再统一返回

每个区间看作一个“限时任务”，在  $\text{left}[i]$  时产生，在  $\text{right}[i]$  时消失，任务代价是  $\text{length}[i]$

每个询问看作一个在  $\text{queries}[i]$  时发生的“询问”事件——哪个可做的任务代价最小？

排序 + 堆 or set 维护

vs. 在线处理询问需要支持区间修改的线段树

# 期末答疑

时间：8月26日 20:00（正常周四上课时间 第21课）

讲解期中考试题目、答疑



# THANKS

 极客时间 | 训练营