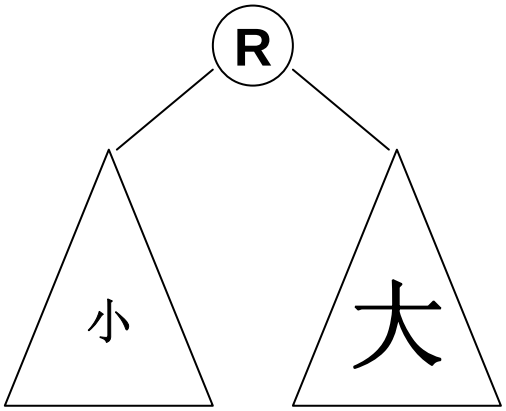
树之 习题选讲

浙江大学 陈 越

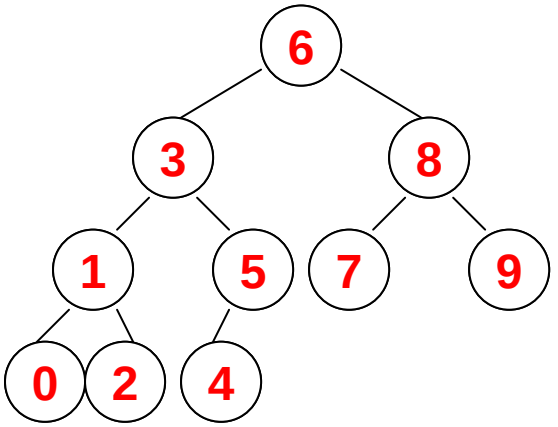
Complete Binary Search Tree
完 全 二 叉 搜 索 树

题意理解

- 二叉搜索树



- 完全二叉树



输入：1 2 3 4 5 6 7 8 9 0

输出：6 3 8 1 5 7 9 0 2 4

树的表示法：链表 vs. 数组

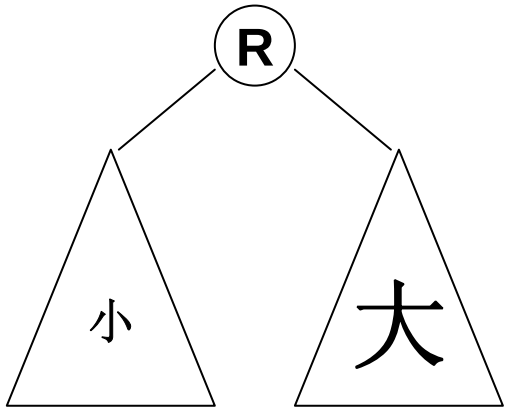
- 需要的操作
  - 填写数字（某种遍历）
  - 层序遍历

  - 完全二叉树，不浪费空间
  - 层序遍历 == 直接顺序输出
- 数组完胜！

核心算法

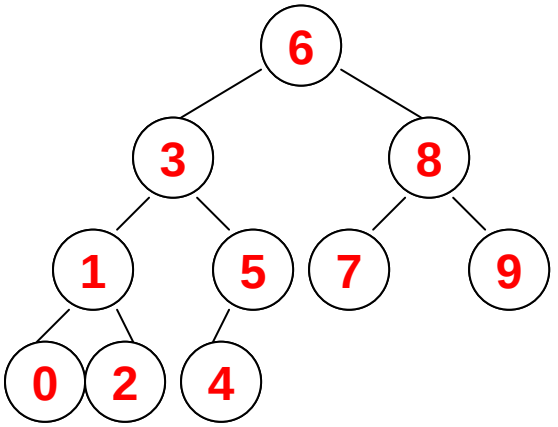
1．排序
2．因为是完全二叉树，所以可以算出左节点和右节点的个数
3．递归

- 二叉搜索树



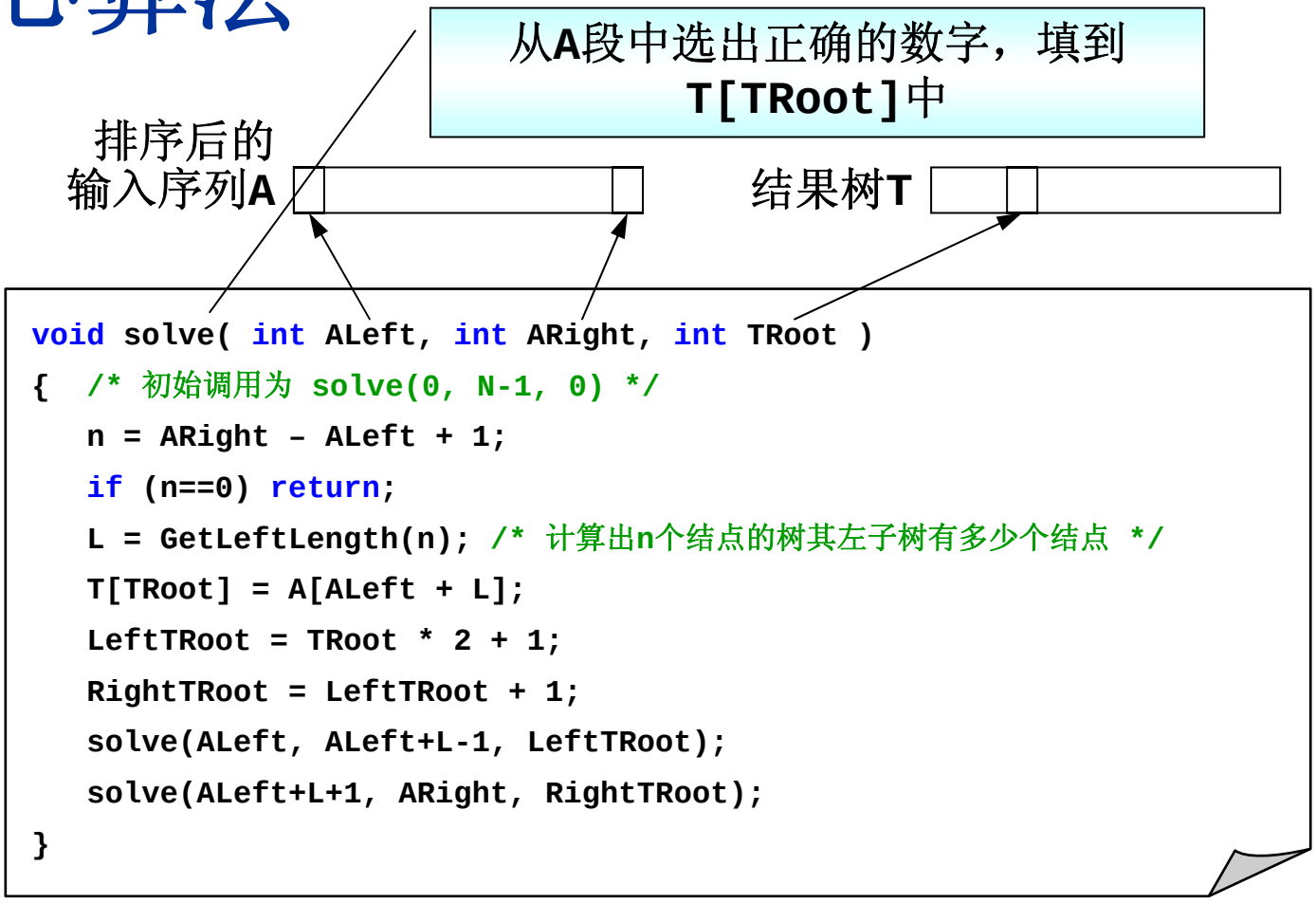
- 完全二叉树

这其实是一个先序遍历的应用



排序：0 1 2 3 4 5 6 7 8 9

核心算法




```
void solve( int ALeft, int ARight, int TRoot )
{  /* 初始调用为 solve(0, N-1, 0) */
   n = ARight – ALeft + 1;
   if (n==0) return;
   L = GetLeftLength(n); /* 计算出n个结点的树其左子树有多少个结点 */
   T[TRoot] = A[ALeft + L];
   LeftTRoot = TRoot * 2 + 1;
   RightTRoot = LeftTRoot + 1;
   solve(ALeft, ALeft+L-1, LeftTRoot);
   solve(ALeft+L+1, ARight, RightTRoot);
}
```

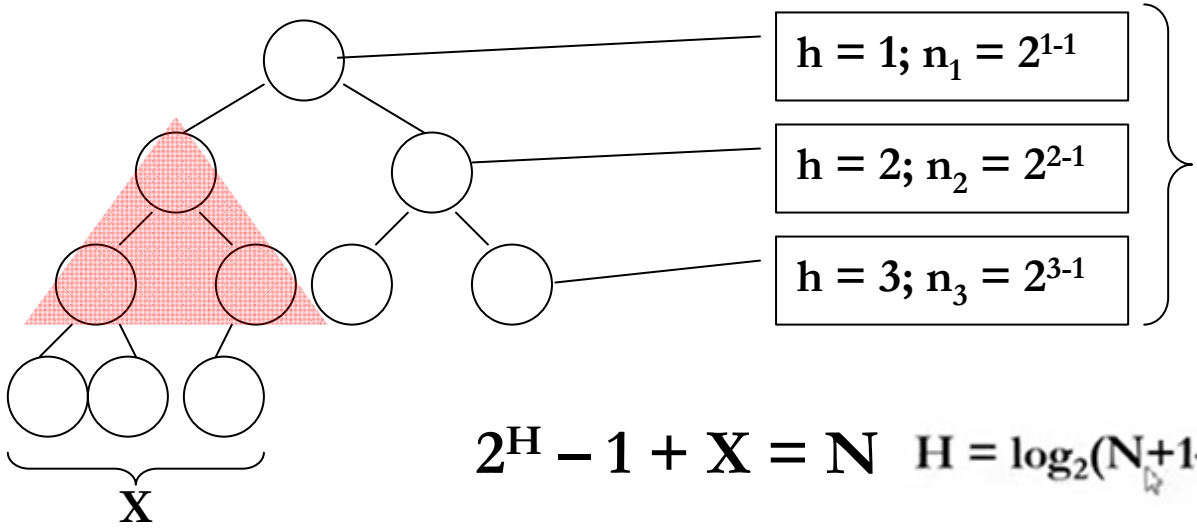
排序

```
int compare(const void*a, const void*b)
{
   return *(int*)a-*(int*)b;
}
```

```
#include <stdlib.h>
int main()
{  ……
   qsort(A, N, sizeof(int), compare);
   ……
}
```

计算左子树的规模

讨论7.1问，为什么最小X不是1？没想明白，复习的时候再思考一下



节点总数：

$2^H - 1$

$2^H - 1 + X = N$ $H = \log_2(N+1-X)$

因为X的大小对结果的影响很小，因此直接忽略掉x，并做向下取整，得到1的公式

➡ $H = \lfloor \log_2(N+1) \rfloor$ ①

这里的最大和最小的X指的是左子树的最下面一层的最大最小取值

左子树的个数：

$L = 2^{H-1} - 1 + X$?

③

最小X = 0

最大X = 2^{H-1}

$X = \min\{ X, 2^{H-1} \}$ ②

自己算一下这个最大X是怎么得到的