

树之 习题选讲

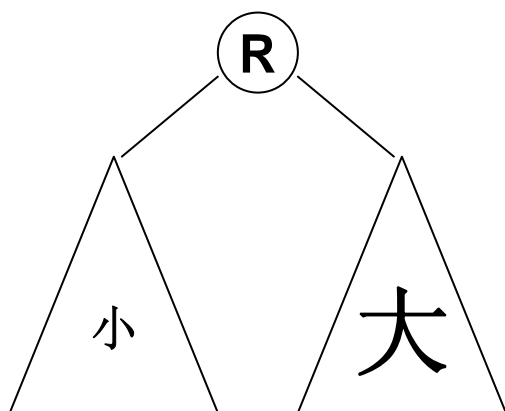
浙江大学 陈 越

Complete Binary Search Tree

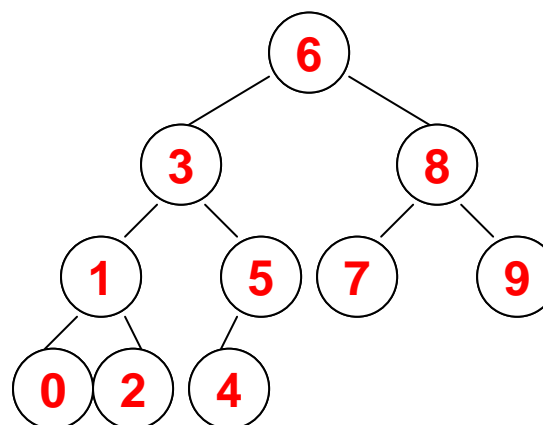
完全二叉搜索树

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

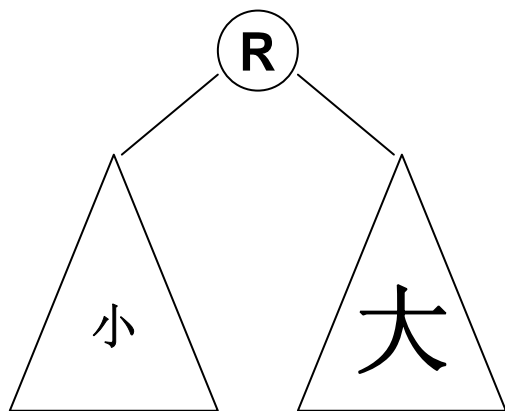
输出: 6 3 8 1 5 7 9 0 2 4

树的表示法：链表 vs. 数组

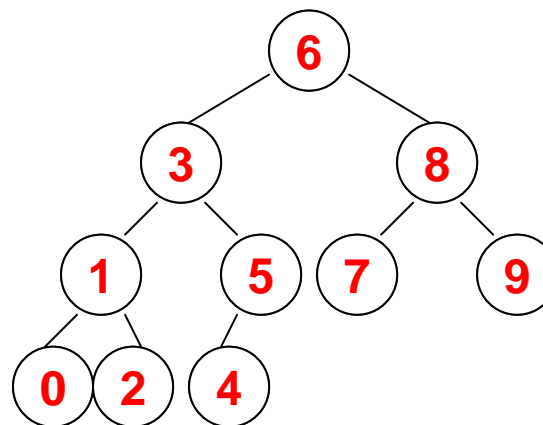
- 需要的操作
 - 填写数字（某种遍历）
 - 层序遍历
 - 完全二叉树，不浪费空间
 - 层序遍历 == 直接顺序输出
- 数组完胜！

核心算法

■ 二叉搜索树



■ 完全二叉树



排序: 0 1 2 3 4 5 6 7 8 9

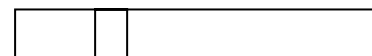
核心算法

排序后的
输入序列A



从A段中选出正确的数字，填到
T[TRoot]中

结果树T



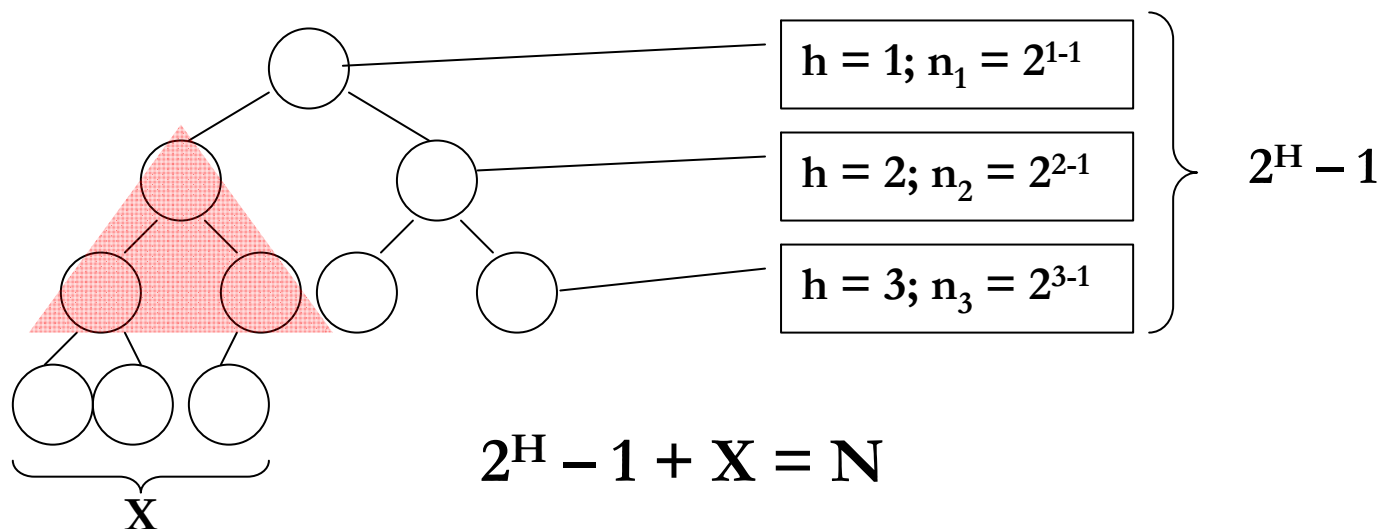
```
void solve( int ALeft, int ARight, int TRoot )
{ /* 初始调用为 solve(0, N-1, 0) */
  n = ARight - ALeft + 1;
  if (n==0) return;
  L = GetLeftLength(n); /* 计算出n个结点的树其左子树有多少个结点 */
  T[TRoot] = A[ALeft + L];
  LeftTRoot = TRoot * 2 + 1;
  RightTRoot = LeftTRoot + 1;
  solve(ALeft, ALeft+L-1, LeftTRoot);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



➡ $H = \lfloor \log_2(N+1) \rfloor$ ①

$$\left. \begin{array}{l} \text{③} \quad L = 2^{H-1} - 1 + X? \\ \text{最小} X = 0 \\ \text{最大} X = 2^{H-1} \end{array} \right\} X = \min\{X, 2^{H-1}\} \quad \text{②}$$