

算法导论 XII 章第 1 节习题

Louis1992

有需要可以[联系我](#)

我的 [github](#) [欢迎大家帮我完成算法导论](#)
[我的博客](#)

May 3, 2015

Exercises 12.1-1 For the set of keys $\{1, 4, 5, 10, 16, 17, 21\}$, draw binary search trees of height 2, 3, 4, 5, and 6.

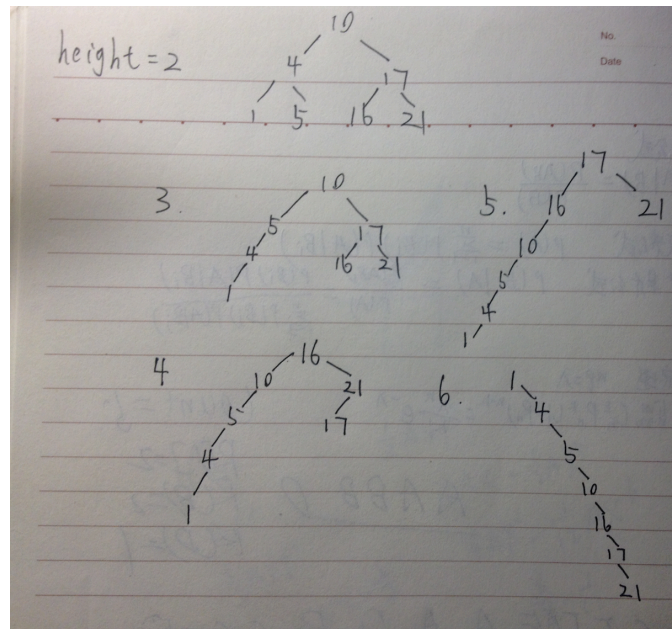


Figure 1: 6.1-1 的答案

Exercises 12.1-2 What is the difference between the binary-search-tree property and

the min-heap property (see page 129)? Can the min-heap property be used to print out the keys of an n -node tree in sorted order in $O(n)$ time? Explain how or why not.
做不到，因为基于比较的排序的上限是 $n \lg n$ 。堆排序建堆时间是 $O(n)$ ，但是每次拿出去后要用 $\lg n$ 的时间去调整。

Exercises 12.1-3 Give a nonrecursive algorithm that performs an inorder tree walk. (Hint: There is an easy solution that uses a stack as an auxiliary data structure and a more complicated but elegant solution that uses no stack but assumes that two pointers can be tested for equality.)

```
void inorderTraversal(TreeNode *root) {
    stack<TreeNode*> s;
    TreeNode *p = root;

    while (p != NULL || !s.empty())
    {
        while (p != NULL)
        {
            s.push(p);
            p = p->left;
        }

        p = s.top();
        s.pop();
        cout << p->val;
        p = p->right;
    }
}
```

Exercises 12.1-4 Give recursive algorithms that perform preorder and postorder tree walks in $\Theta(n)$ time on a tree of n nodes.

```
void preorder(TreeNode *root) {

    if (!root) return;
```

```

        cout << root->val << endl;
        preorder(root->left);
        preorder(root->right);
    }

```

```

void postorder(TreeNode *root) {

    if(!root) return;

    postorder(root->left);
    postorder(root->right);
    cout << root->val << endl;

}

```

Exercises 12.1-5 Argue that since sorting n elements takes $\Omega(n \lg n)$ time in the worst case in the comparison model, any comparison-based algorithm for constructing a binary search tree from an arbitrary list of n elements takes $\Omega(n \lg n)$ time in the worst case.

如果下界不是 $n \lg n$, 就会产生矛盾。我们之前知道可以用 $O(n)$ 的时间按序遍历输出 BST 中的元素, 如果基于比较的 BST 构造下界不是 $n \lg n$, 那我们就可以利用这个方法产生基于比较的速度优于 $n \lg n$ 的排序算法, 矛盾。