

一、单选、多选或填空题（10 小题，每空 1 分，共 10 分）

1. 12。
2. BC
3. (10, 7, -9, 0, 47, 23, 1, 8, 98, 36) 。
4. CDF
5. 10; 1
6. 198
7. 2080; 251

得分

二、简答辨析题（6 小题，共 17 分）

1. 【3 分】答案：

- (1) 0 为根结点；1, 3 为 0 的孩子结点；2, 4 为 1 的孩子结点。
- (2) 结点 0~4 所对应的区间依次为：[1, 5]; [1, 3]; [1, 1]; [4, 4]; [2, 2]
- (3) 是。设 $I_u = [i_u, j_u]$, $I_v = [i_v, j_v]$ ，则 $I_u \supseteq I_v \Leftrightarrow i_u \leq i_v, j_u \geq j_v$ 。 $j_u \geq j_v$ 说明在树的后序遍历中，先访问 v 、后访问 u ，这说明 u 要么是 v 的祖先结点，要么位于 v 的右侧。而 $i_u \leq i_v$ 说明 u 不在 v 的右侧。因此 u 是 v 的祖先结点， u 与 v 之间存在所有边都在树中的路径。

2. 【2 分】答案：

(1) (1 分) 最优排序方法为冒泡排序或选择排序。每一趟可以找到 1 个当前最大元素， k 趟之后终止。每一趟的比较次数分别为 $n-1, n-2, \dots, n-k$ 。因此，总比较次数约为 100000 次。

(2) (2 分) 建立最大堆，最多需要 $2n$ 次比较：建堆过程，所有非叶子节点下移步数总和最多为 n ，每次下移进行 2 次比较，所以建堆比较次数为 $2n$ 。之后每次 `pop_max` 调整堆，根结点最多下移 $\log n$ ，每次下移比较次数也为 2 次。因此，总比较次数为 $2*(n+k\log n)=20280$ 次。

注：

(1) 中，写出最优时应使用的排序算法，即使计算错误，可以给部分分数。

(2) 中，可以堆每一部分计算给部分分数。

3. 【2 分】

(1)

参考答案：

(1).根据题意，散列表长度为 $L = 7/0.7 = 10$ ；因此此题需要构建的哈希表是下标为 0~9 的一维数组。

构造的散列表为：

地址	0	1	2	3	4	5	6	7	8	9
关键字	7	14		30	9	8	11	18		

(2) 等概率情况下查找成功平均查找长度：

根据 (1) 的构造过程，查找次数如下表所示：

Key	7	8	30	11	18	9	14
Count	1	1	1	1	2	2	2

所以 $ASL_{success} = (1+1+1+1+2+2+2) / 7 = 10/7$ 。

等概率情况下，查找 0~6 位置查找不成功的次数表如下表所示

Key	0	1	2	3	4	5	6
Count	3	2	1	3	4	4	5

所以 $ASL_{unsuccess} = (3+2+1+3+4+4+5) / 7 = 22/7$ 。

4. 【2 分】

- 在一次检索中涉及的节点最多的情况即从树的根部检索到深度最大的叶节点处。该情况下，涉及的节点个数等于树的高度。一颗包含 n 个关键码的 m 阶 B-树，其最大高度为 $h = \log_{\lceil m/2 \rceil} (\frac{n+1}{2}) + 1$ 。(推导过程如下)

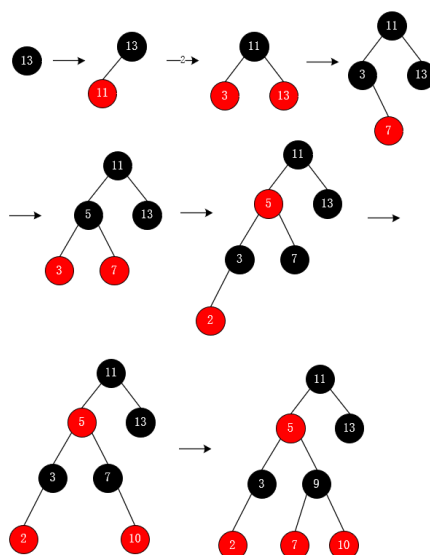
$$\begin{aligned} n &\geq 1 * 1 + 2 * (\lceil m/2 \rceil^0 + \lceil m/2 \rceil^1 + \lceil m/2 \rceil^2 + \dots + \lceil m/2 \rceil^{h-2}) * (\lceil m/2 \rceil - 1) \\ &\geq 1 + 2 * (\frac{1 - \lceil m/2 \rceil^{h-1}}{\lceil m/2 \rceil - 1}) * (\lceil m/2 \rceil - 1) \\ &\geq 1 + 2 * (\lceil m/2 \rceil^{h-1} - 1) \end{aligned}$$

$$\begin{aligned} \lceil m/2 \rceil^{h-1} &\leq \frac{n+1}{2} \\ h &\leq \log_{\lceil m/2 \rceil} (\frac{n+1}{2}) + 1 \end{aligned}$$

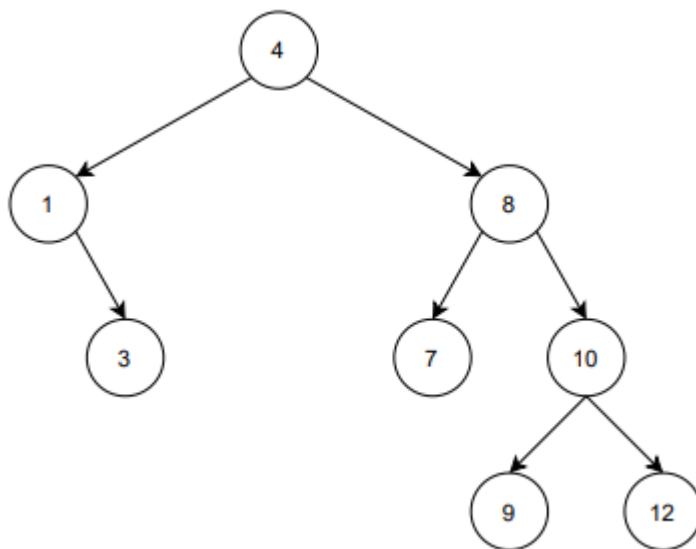
- 在一次检索中最少只涉及到一个结点。

5. 【2 分】

参考答案：

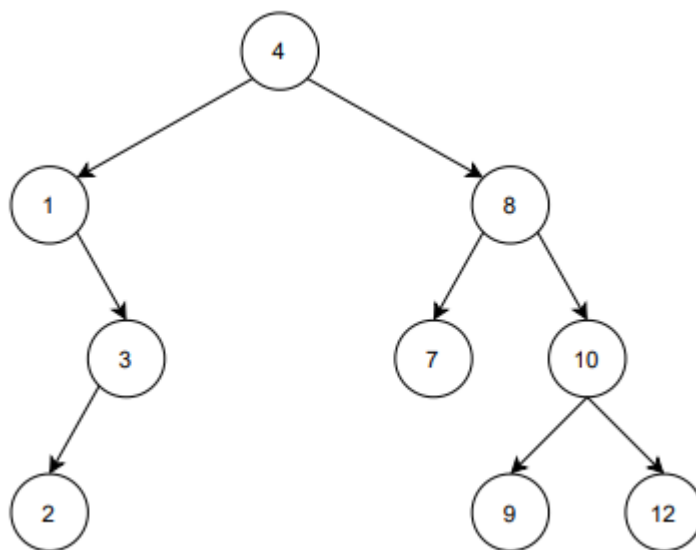


6. 【2 分】

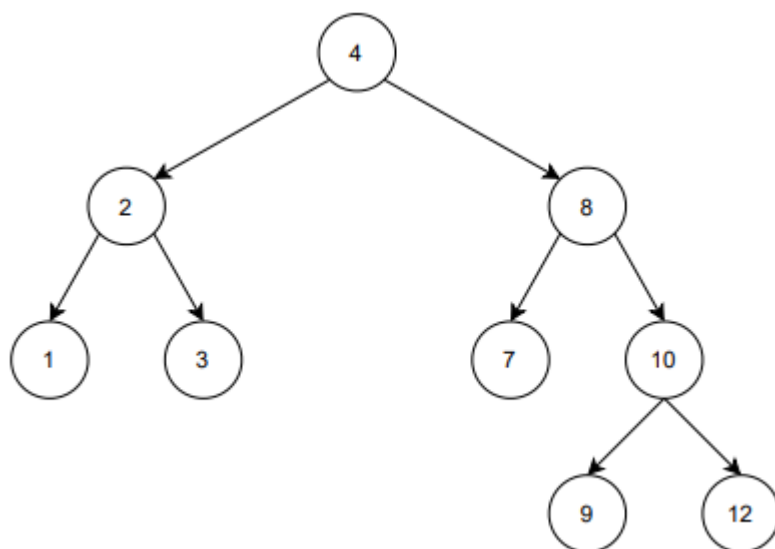


$\log_2 n$, 每次插入后得到的结果如下图：

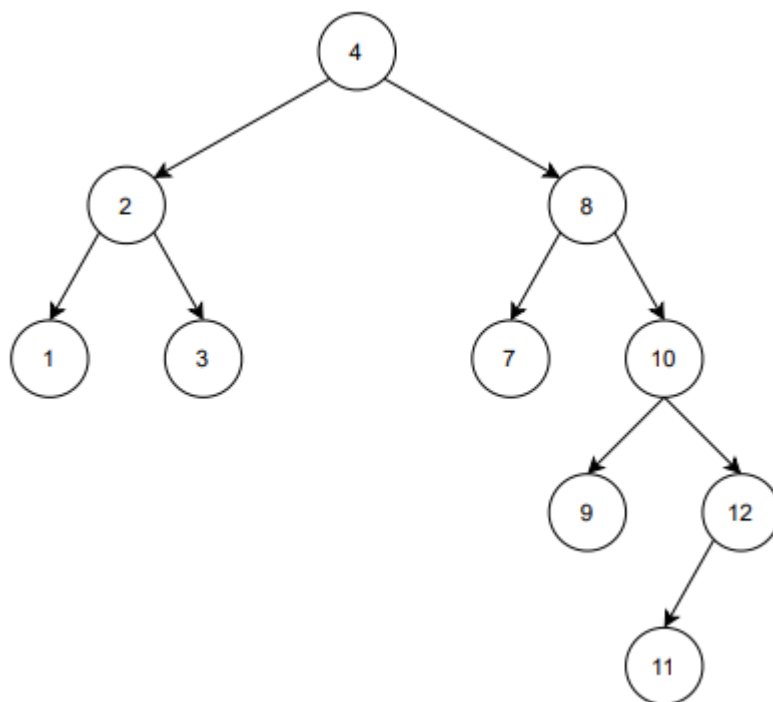
插入 2:



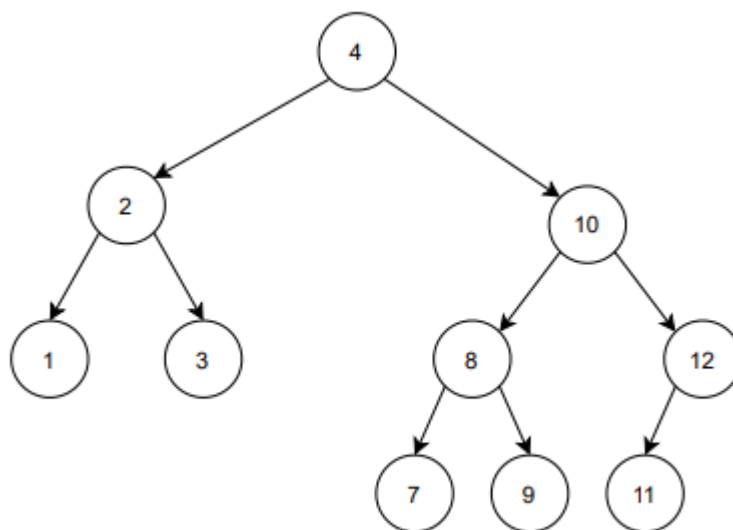
先右旋再左旋：



插入 11：



左旋：



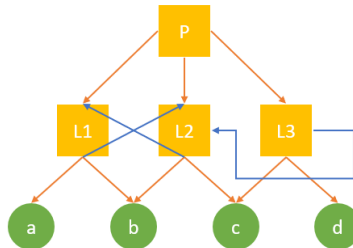
(指出如何旋转占一分，每次插入结束的树占一分)

7. 【3 分】答案：

(1) 如图：要注意 child 的顺序

(2) 正无穷

(3) `head(tail(head(tail(tail(P)))))`



得分

三、算法填空题（每空 1 分，共 5 分）

答案：

- (1) `temp[index1] <= temp[index2]`
- (2) `array[k] = temp[index2--];`
- (3) `result += mid - index1 + 1;`
- (4) `count_swap_left_right(array, left, right);`
- (5) `l_result + r_result + lr_result;`

解析：本题 3 个考点

1. 冒泡排序每次交换的是 2 个相邻元素，消除 1 个逆序对。因此本题实际上是求数组 `array` 中的逆序对个数；
2. 采用基于归并排序的方式，用 $O(n \log n)$ 算法求逆序对；
3. 归并排序实现的优化：归并 2 个数组时，第 1 个数组顺序复制到临时数组中，第 2 个数组逆序复制，然后从临时数组两端往中间遍历。

得分

四、设计分析题（2 小题，共 8 分）

注：本部分答题应申明算法思想，进行必要的算法复杂性分析，代码加以恰当注释。

1. 【5 分】

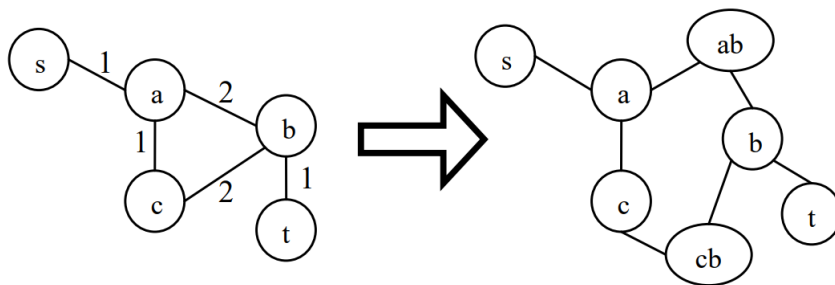
答案

方法 1：使用 `dijkstra` 算法直接求解得到。最佳方案为删除旧值的最小堆实现，

时间复杂度 $O((m+n)\log n)$ 。若使用不删除旧值的最小堆，时间复杂度为 $O((m+n)\log m)$ ，在稀疏图下复杂度可能会更高。若直接扫描数组，时间复杂度 $O(n^2)$ ，复杂度最高。

方法 2：将 G 通过如下操作转换为无权图 $G'=(V', E')$ ：

- 1) $V'=V, E'=E$
- 2) 对每条权为 2 的边 $(u, v) \in E$ ：
 - a) 将点 uv 加入 V'
 - b) 从 E' 中移去 (u, v)
 - c) 将 $(u, uv), (uv, v)$ 加入 E' 中



从而在 G 中找权重和最短的路径等价于在 G' 中找最短路径。针对无权最短路径问题，可以使用 BFS 解决。 G' 至多有 $m+n$ 个点， $2m$ 条边。转换过程和 BFS 的时间复杂度都为 $O(m+n)$ ，所以总的时间复杂度也为 $O(m+n)$ 。

在稀疏图上，方法 2 比方法 1 更优。

2. 【4 分】

解：

- 1) 将 $m-1$ 结点 Splay 到根，第 $n+1$ 结点 Splay 到根的右儿子，将其左儿子赋值为 0；
- 2) 将 $m-1$ 结点 Splay 到根，第 $n+1$ 结点 Splay 到根的右儿子，递归翻转左子树即可。