

中国科学院大学

人工智能学院

课程编号: B2511006Y

课程名称: 数据结构与算法

任课教师: 董未名

一、选择题（共 60 分）

1. 当输入非法时，一个“好”的算法会进行适当处理，而不是产生难以理解的输出结果。这称为算法的（ B ）。
A. 可读性 B. 健壮性 C. 正确性 D. 有穷性
2. 在一个初始为空的向量上依次执行：insert(0, 2), insert(1, 6), put(0, 1), remove(1), insert(0, 8) 后的结果是（ C ）。
A. {6, 2, 8} B. {2, 6, 0, 8} C. {8, 1} D. {2, 1, 8}
3. 对于插入过程排序中的已排序子序列（设其长度为 k），下列说法正确的是（ C ）。
A. 其中的元素是整个序列中最小的 k 个元素
B. 其中的元素是整个序列中最大的 k 个元素
C. 其中的元素是原序列中位于前方的 k 个元素
D. 其中的元素是原序列中位于后方的 k 个元素
4. 选择排序算法的哪种实现是稳定的：（ A ）
A. 每一趟将最小元素移到前方，对于多个相等的最小元素，选取其中位置最靠前者。
B. 每一趟将最大元素移到后方，对于多个相等的最大元素，选取其中位置最靠前者
C. 每一趟将最小元素移到前方，对于多个相等的最小元素，选取其中位置最靠后者。
D. 以上实现皆稳定。
5. 利用栈结构进行中缀表达式求值，什么时候进行实际的运算？（ D ）
A. 每遇到一个新的操作数
B. 每遇到一个新的操作符
C. 当前的操作符比栈顶的操作符优先级高
D. 当前的操作符比栈顶的操作符优先级低
6. 分别采用每次追加固定内存空间和每次内存空间翻倍两种扩容策略，规模为 n 的向

量插入元素的分摊时间复杂度分别为：（ A ）

A. $O(n), O(1)$ B. $O(n), O(n)$ C. $O(1), O(1)$ D. $O(n), O(\log 2n)$

7. 给定序列 $\{1, 2, 3 \dots i \dots j \dots k \dots n\}$ 。下列哪个序列一定不是该序列的栈混洗：（ C ）

A. $\{\dots i \dots j \dots k \dots\}$ B. $\{\dots k \dots j \dots i \dots\}$ C. $\{\dots k \dots i \dots j \dots\}$ D. $\{\dots j \dots k \dots i \dots\}$

8. $V = \{1, 2, 3, 4, 5, 6, 7\}$ ，在 V 中用 Fibonacci 查找元素 1，被选取为轴点 mi 的元素依次是（ D ）

A. 4, 3, 2, 1 B. 4, 2, 1 C. 5, 2, 1 D. 5, 3, 2, 1

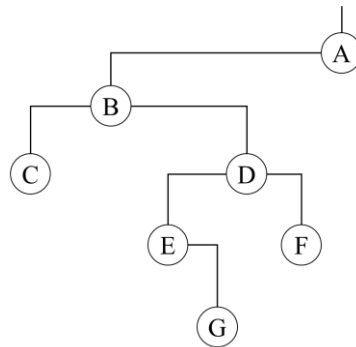
9. 用父节点+孩子节点的方法存储 n 个节点的树,需要的空间是（ B ）。

A. $O(1)$ B. $O(n)$ C. $O(n \log n)$ D. $O(n^2)$

10. 二叉树的中序遍历中第一个被访问的节点是（ A ）。

A. 最左的节点 B. 最右的节点 C. 根节点 D. 左侧分枝的叶节点

11. 对下图二叉树进行层次遍历,节点 F 正欲出队时队列中的元素从队头到队尾为（ B ）。



A. F B. F, G C. E, F D. E, F, G

12. 下列关于树的命题中错误的是：（ D ）

- A. 顶点数为 n 的树的边数为 $n-1$ 。
- B. 树中任意两顶点之间存在唯一路径。
- C. 在树中添加任一条边都会破坏树的结构。
- D. 在树中删除任一条边得到的还是树。

13. 关于二叉树遍历序列之间关系的说法错误的是：（ D ）

- A. 已知先序遍历序列和中序遍历序列可以确定后序遍历序列
- B. 已知中序遍历序列和后序遍历序列可以确定先序遍历序列
- C. 已知中序遍历序列和后序遍历序列可以确定层次遍历序列
- D. 已知先序遍历序列和后序遍历序列可以确定中序遍历序列

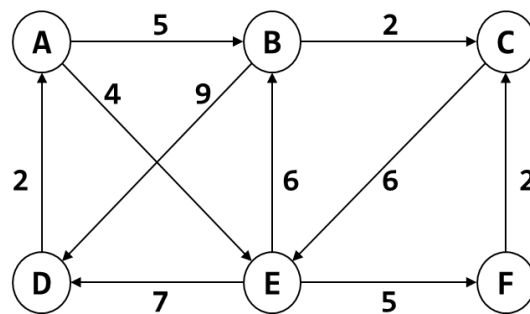
14. 对平衡二叉树进行插入操作，对待插入的目标元素 e 进行查找后，若查找失败， $_hot$ 指向的节点为：（ B ）

- A. 待插入的节点 B. 被插入后的父亲 C. 被插入后的左孩子 D. 根节点

15. AVL 树中插入节点引发失衡，经旋转调整后重新平衡，此时包含节点 g, p, v 的子树高度（ B ）

- A. 减小 1 B. 不变 C. 增加 1 D. 有可能不变也有可能增加 1

16. 使用 Dijkstra 算法求下图中从顶点 A 到其他各顶点的最短路径，依次得到的各最短路径的目标顶点是（ A ）



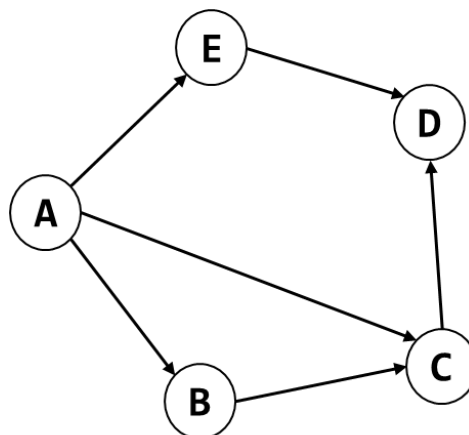
- A. {E, B, C, F, D} B. {E, B, F, C, D} C. {E, B, D, C, F} D. {E, B, C, D, F}

17. B 树高度的减少只会发生于（ A ）

- A. 根节点的两个孩子合并 B. 根节点被删除
C. 根节点发生旋转 D. 根节点有多个关键码

18. 对下图进行拓扑排序，可以得到的不同的拓扑序列的个数是（ B ）

- A. 4 B. 3 C. 2 D. 1



19. 伸展树每次访问过某节点后都会把该节点（ C ）

- A. 删除 B. 上移一层 C. 移动到根节点 D. 再次访问该节点

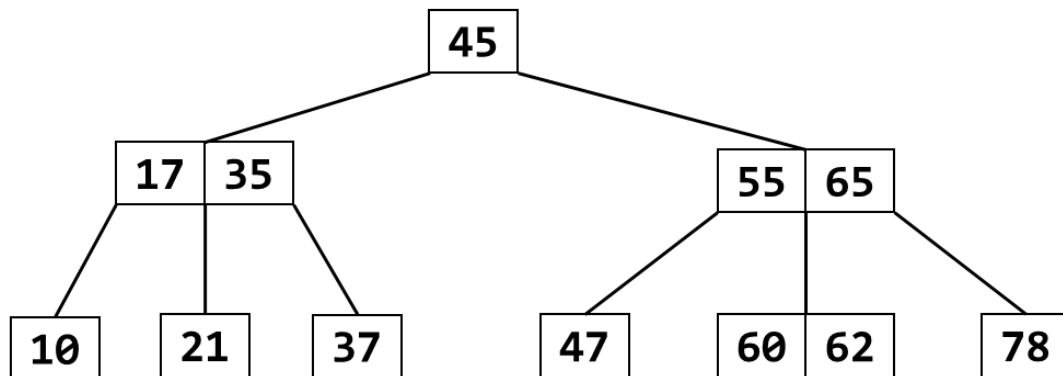
20. G 是有向无环图, (u, v) 是 G 中的一条由 u 指向 v 的边。对 G 进行 DFS 的结果是:
(C)
- A. $dTime(u) > dTime(v)$ B. $dTime(u) < dTime(v)$
C. $fTime(u) > fTime(v)$ D. $fTime(u) < fTime(v)$
21. 文本串的长度为 n , 模式串的长度为 m , 蛮力匹配的时间复杂度为 (D)
- A. $O(m)$ B. $O(n)$ C. $O(m \log n)$ D. $O(mn)$
22. 对序列 $A[0, n)$ 用快速排序算法进行排序, u 和 v 是该序列中的两个元素。在排序过程中, u 和 v 发生过比较, 当且仅当 (假定所有元素互异): (C)
- A. $u < v$
B. u 在某次被选取为轴点
C. 对于所有介于 u 和 v 之间的元素 (包括 u 和 v 本身), 它们之中第一个被选为轴点的是 u 或者 v
D. 所有比 u 和 v 都小的元素都始终没有被选为轴点
23. 允许对队列进行的操作是 (C)
- A. 对队列中的元素排序
B. 取出最近进队的元素
C. 删除队头元素
D. 在队头元素之前插入元素
24. 在使用独立链法解决冲突的散列表中查找某关键码, 经过一系列的试探, 最终查找成功。这些试探的元素 (B)。
- A. 关键码的散列地址均不相同
B. 关键码的散列地址一定相同
C. 关键码的散列地址可能不同
D. 无法判断
25. 已知运算符包括 $+$ 、 $-$ 、 $*$ 、 $/$ 、(和), 将中缀表达式 $3+2-6*((5+1)/2-3)+9$ 转换为逆波兰表达式时, 符号栈最少应该能保存多少个符号? (A)
- A. 5 B. 6 C. 7 D. 8
26. 若元素 a 、 b 、 c 、 d 、 e 、 f 依次入栈, 允许入栈、出栈操作交替进行, 但不允许连续 3 次进行出栈操作, 则不可能得到的出栈序列是 (D)
- A. d, c, e, b, f, a B. c, b, d, a, e, f
C. b, c, a, e, f, d D. a, f, e, d, c, b

27. 若平衡二叉树的高度为 6，且所有非叶结点的平衡因子均为 1，则该平衡二叉树的结点总数为（ B ）

- A. 12 B. 20 C. 32 D. 33

28. 已知一棵 3 阶 B 树，如下图所示。删除关键码 78 后得到一棵新的 B 树，其最右叶子结点中的关键码是（ D ）

- A. 60 B. 60、62 C. 62、65 D. 65



29. 下列选项给出的是从根分别到达两个叶结点路径上的权值序列，能属于同一棵哈夫曼树的是（ D ）

- A. 24, 10, 5 和 24, 10, 7 B. 24, 10, 5 和 24, 12, 7
C. 24, 10, 10 和 24, 14, 11 D. 24, 10, 5 和 24, 14, 6

30. 下列关于生成树的说法正确的是（ C ）。

- A. 一个图的最小生成树一定不唯一
B. 若图有 3 个最小生成树 T1、T2、T3，并且在 T1 中权值最小的边为 e ，则 e 一定会出现在 T2 和 T3 中
C. 某个无向图存在相同权值的边，它的最小生成树仍然可能是唯一的
D. 一个图一定有大于等于一棵最小生成树

二、综合应用题（共 40 分）

1. （8 分）本题关于堆排序，请在空白处填入对应的关键语句。

```
template <typename T> void Vector<T>::heapSort( Rank lo, Rank hi ) {
    T* A = _elem + lo; Rank n = hi - lo; heapify( A , n );
    while ( 0 < --n )
        { swap( A[0], A[n] ); percolateDown( A, n, 0 ); }
}

template <typename T> void heapify( T* A, const Rank n ) {
    for ( Rank i = n / 2 - 1; - 1 != i; i-- )
        percolateDown( A, n, i );
}

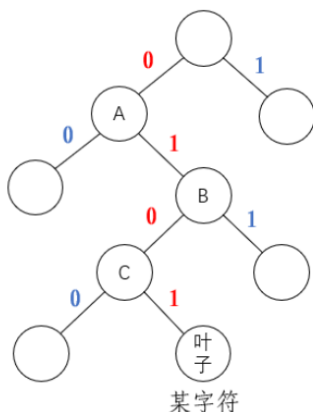
template <typename T> Rank percolateDown( T* A, Rank n, Rank i ) {
    Rank j;
    while ( i != ( j = ProperParent( A, n, i ) ) )
        swap( A[i], A[j] ), i = j;
    return i;
}
```

2. （10 分）本题关于 Huffman 编码。对于一个字符集中具有不同权值的字符进行 Huffman 编码时，如果已知某个字符的 Huffman 编码为 0101，对于其他无字符的 Huffman 编码，请分析说明：

- （6 分）具有哪些特征的编码是不可能的
- （4 分）具有哪些特征的编码是一定会有的

解：

已知某字符的 Huffman 编码为 0101，因此该字符在 Huffman 树中的位置如下：



假设结点 A、B、C 的位置如左图：

在哈夫曼树中，所有字符一定位于叶子结点上，因此图中各分支节点所对应的编码是不可能出现的，即：0，01，010，三个编码不可能出现；（即结点 A、B、C 不可能是叶子结点）

另外，叶子结点的下层不可能再出现其他结点，因此以 0101 为前缀的编码也不可能出现。

注意哈夫曼树的特性——所有分支节点的度必然是 2。（不存在度为 1 的结点）

- a. 0、01、010 三个编码不可能出现（各 1 分）；以 0101 为前缀的编码也不可能出现（2 分）。
 b. 以 1、00、011、0100 为开头的编码一定会有（各 1 分）。
 无分析说明文字扣 3 分，多写一个错误的编码扣 0.5 分，最多扣 10 分。

3. （12 分）采用散列函数 $H(k) = 3 * k \bmod 13$ ，并用线性探测开放地址法处理冲突，在数列地址空间 $[0 \dots 12]$ 中对关键码序列 22, 41, 53, 46, 30, 13, 1, 67, 51。

- a. （3 分）构造散列表（画示意图，类似下表）。

散列地址
关键码

- b. （3 分）计算装填因子。
 c. （3 分）计算成功的平均查找长度。
 d. （3 分）计算不成功的平均查找长度。

解：

- a. 如下图所示：（在数列地址空间 $[0..12]$ 中，此句意思为：表长 13）

构造的哈希表如下：

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	13	22		53	1		41	67	46		51		30

$$H(22) = 3 * 22 \bmod 13 = 1$$

$$H(41) = 3 * 41 \bmod 13 = 6$$

$$H(53) = 3 * 53 \bmod 13 = 3$$

$$H(46) = 3 * 46 \bmod 13 = 8$$

$$H(30) = 3 * 30 \bmod 13 = 12$$

$$H(13) = 3 * 13 \bmod 13 = 0$$

$$H(1) = 3 * 1 \bmod 13 = 3$$

$$H(67) = 3 * 67 \bmod 13 = 6$$

$$H(51) = 3 * 51 \bmod 13 = 10$$

装填因子 = 表中填入的记录数 / 散列表的长度

$$= 9 / 13$$

装填因子越小，发生冲突的可能性就越小。

假如设定散列函数是“均匀”的，则影响平均查找长度的因素只有两个：处理冲突的方法和装填因子。

- b. 散列表的地址空间大小为 13，有 9 个关键字已经插入，因此装填因子 = $9/13$
 c.

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	13	22		53	1		41	67	46		51		30
查找成功时的查找长度	1	1		1	2		1	2	1		1		1
查找失败时的查找长度	3	2	1	3	2	1	4	3	2	1	2	1	4

因此查找成功时的平均查找长度 $ASL_{成功} = (1 \times 7 + 2 \times 2) / 9 = 11/9$

查找失败时的平均查找长度 $ASL_{失败} = (3+2+1+3+2+1+4+3+2+1+2+1+4) / 13 = 29/13$

查找失败（注意）

对于散列地址为2的关键字，查找失败时的查找长度为1。（刚开始检查的Hash地址就是空的，可以直接确定查找失败）

对于散列地址为12的关键字，查找失败时的查找长度为4。（发生冲突时，通过线性探测，找到第一个空，也就是2的位置，才能确定查找失败）

d. 查找失败时的平均查找长度 $ASL_{失败} = (3+2+1+3+2+1+4+3+2+1+2+1+4) / 13 = 29/13 = 2.23$

例如：

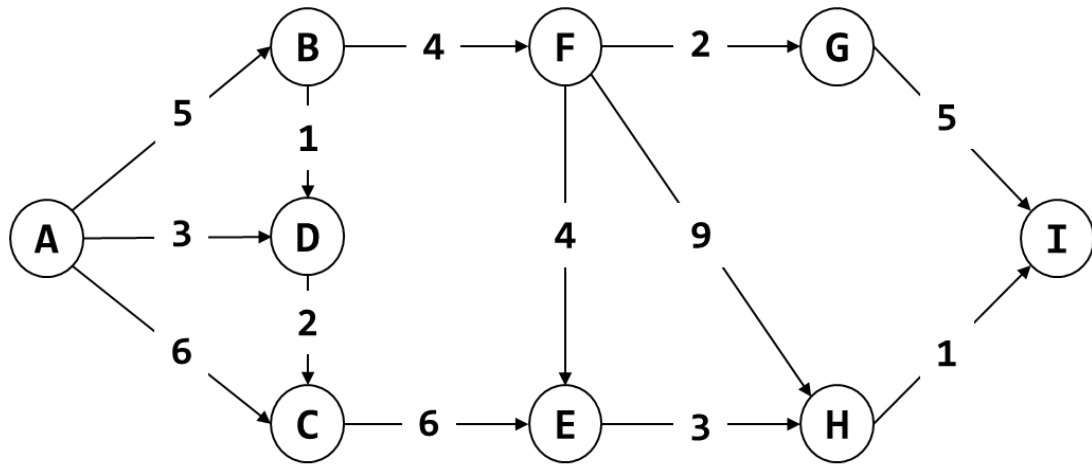
对于散列地址为0的关键字，查找失败时的查找长度为3。（发生冲突时，通过线性探测，找到第一个空,也就是2的位置，才能确定查找失败）

对于散列地址为1的关键字，查找失败时的查找长度为2。（发生冲突时，通过线性探测，找到第一个空，也就是2的位置，才能确定查找失败）

对于散列地址为2的关键字，查找失败时的查找长度为1。（刚开始检查的Hash地址就是空的，可以直接确定查找失败）

4. （10分）本题关于图结构。下图中的AOE网表示一个工程。请根据AOE网的相关概念和该图回答下列问题。

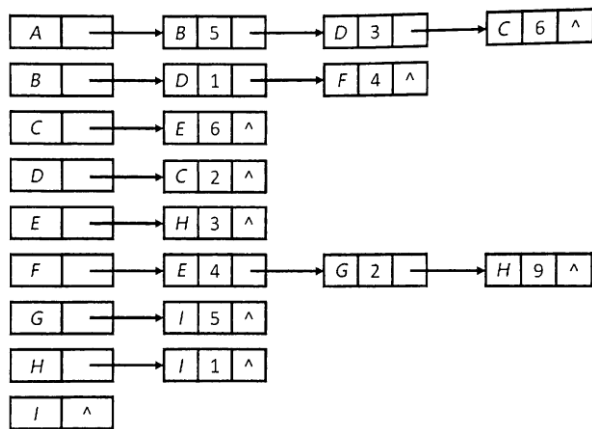
通过有向图的边表示活动的网称为AOE网（Activity On Edge Network）。AOE网用边表示活动，用顶点表示事件（活动的完成）。边是带权的，表示活动需要的时间。AOE网中，入度为0的点表示一个工程的开始（源点），出度为0的点表示一个工程的结束（汇点）。在AOE网中，从源点到汇点最长的路径称为关键路径，在关键路径上的活动称为关键活动。AOE网中的活动是可以并行进行的，只有在某顶点所代表的事件发生后，从该顶点出发的各有向边所代表的活动才能开始。只有在进入每顶点的各有向边所代表的活动都已经结束后，该顶点所代表的事件才能发生。



- (2 分) 画出该图的邻接表。
- (2 分) 计算此工程的最少所需时间。
- (2 分) 给出该 AOE 网的关键路径。
- (4 分) 分析加速哪些活动可以缩短工程的所需时间。

解:

- a. 该图的邻接表如图所示。



- 最少所需时间为 19。
- 关键路径为 A-B-F-H-I。
- 加速 A-B、B-F、F-H、H-I 均可以缩短工程所需时间 (各 1 分)。