

1. 采用线性探测法处理散列时的冲突,当从哈希表删除一个记录时,不应将这个记录的所在位置置空,因为这会影响以后的查找()对
2. 以下哪个不属于单向哈希表的特征 () B
 - 它把任意长度的信息转换成固定的长度输出
 - 它把固定的信息转换成任意长度信息输出
 - 根据特定的哈希值, 它可以找到对应的原信息值
 - 不同的信息很难产生一样的哈希值
3. 将 10 个元素散列到 100000 个单元的哈希表中,则()产生冲突 可能会
4. 哈希函数有一个共同的性质,即函数值应当以()取其值域的每个值。同等概率
5. HASH 函数冲突处理方式不包括以下哪一项: C
 - 开放定址法
 - 链地址法
 - 插入排序法
 - 公共溢出区发
6. 哈希表的平均查找长度与处理冲突的方法无关() 错
7. 哈希函数的选取平方取中法最好() 错
8. 在哈希查找中,"比较"操作一般也是不可避免的()对
9. 若装填因子 α 为 1,则向哈希表中散列元素时一定会产生冲突() 错
10. 假设把整数关键码 K 散列到有 N 个槽的散列表, 以下那些散列函数是好的散列函数? C
 - $h(k)=k/n$
 - $h(k)=1$
 - $h(k)=k \bmod N$
 - $h(k)=(k + \text{Random}(N)) \bmod N$; $\text{Random}(N)$ 返回一个 0 到 $N-1$ 的整数
11. 哈希表的结点中只包含数据元素自身的信息,不包含任何指针() 错 (拉链法解决冲突时, 需要指针)
12. 执行()操作时,需要使用队列做辅助存储空间 B
 - 查找哈希(Hash)表
 - 广度优先搜索网
 - 前序(根)遍历二叉树
 - 深度优先搜索网
13. 下列不属于 hash 碰撞解决方法的是 ()。单旋转法
 - 线性探测 单旋转法 二次探测
 - 拉链法 双重散列 多重散列
14. 散列文件使用散列函数将记录的关键字值计算转化为记录的存放地址。由于散列函数不是一对一的关系, 所以选择好的 () 方法是散列文件的关键。散列函数和冲突处理

15. 下面有关数据结构的说法是正确的？数组的插入和删除可以 $O(1)$
16. 采用哈希表组织 100 万条记录，以支持字段 A 快速查找，则（ ）拉链式哈希曼最坏查找时间复杂度是 $O(n)$
17. 散列函数有一个共同性质，即函数值应按（ ）取其值域的每一个值。同等概率
18. 已知有一个关键字序列：(19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79) 散列存储在一个哈希表中，若散列函数为 $H(key) = key \% 7$ ，并采用链地址法来解决冲突，则在等概率情况下查找成功的平均查找长度为（ ）。1.5
19. 设有一组记录的关键字为{19,14,23,1,68,20,84,27,55,11,10,79},用链地址法构造哈希表,哈希函数为 $H(key)=key \text{ MOD } 13$,哈希地址为 1 的链中有()个记录 4
20. 下列关于线性表，二叉平衡树，哈希表存储数据的优劣描述错误的是？ D
 - A 哈希表是一个在时间和空间上做出权衡的经典例子。如果没有内存限制，那么可以直接将键作为数组的索引。那么所有的查找时间复杂度为 $O(1)$ ；
 - B 线性表实现相对比较简单
 - C 平衡二叉树的各项操作的时间复杂度为 $O(\log n)$
 - D 平衡二叉树的插入节点比较快
21. 设哈希表长度为 11, 哈希函数 $H(K)=(K \text{ 的第一个字母在字母表中的序号}) \text{ MOD } 11$, 若输入顺序为 (D, BA, TN, M, CI, I, K, X, TA), 采用内散列表, 处理冲突方法为线性探测法, 要求构造哈希表, 在等概率情况下查找成功平均查找长度为() 20/9
22. 采用开放定址法处理散列表的冲突时，其平均查找长度？高于链接法处理冲突
23. 已知一个线性表 (38, 25, 74, 63, 52, 48)，假定采用散列函数 $h(key) = key \% 7$ 计算散列地址，并散列存储在散列表 A【0...6】中，若采用线性探测方法解决冲突，则在该散列表上进行等概率成功查找的平均查找长度为 2.0
24. 以下那种结构，平均来讲获取任意一个指定值最快？() 哈希表
25. 稀疏矩阵压缩的存储方法是：（ ）三元组，十字链表
26. 下列哪两个数据结构，同时具有较高的查找和删除性能？（ ）AVL 数 Hash 表
27. 线程安全的 map 在 JDK 1.5 及其更高版本环境 有哪几种方法可以实现? CD
 - Map map = new HashMap()
 - Map map = new TreeMap()
 - Map map = new ConcurrentHashMap();
 - Map map = Collections.synchronizedMap(new HashMap());
28. 哪个是正确的？ D
 - 对于 unordered_map 和 map 这两个容器，迭代器的有效性皆不受删除操作影响
 - 对于 unordered_map 和 map 这两个容器，迭代器的有效性皆不受插入操作影响
 - 为了保证代码的异常安全性，应该避免在构造函数中抛异常
 - 为了保证代码的异常安全性，应该避免在析构函数中抛异常
29. 为了方便在线性结构的数据中连续插入新的元素，则其数据结构最宜采用（ ） D
 - 顺序存储
 - 散列存储
 - 索引存储
 - 链式存储

30. 10 个数放入模 10hash 链表，最大长度是多少？ 10
30. 以下与数据的存储结构无关的术语是() D
循环队列 链表 哈希表 栈
31. 在哈希查找中,"比较"操作一般也是不可避免的() 对
32. 负载因子(装填因子)是哈希表的一个重要参数,它反映哈希表的装满程度() 对
33. hash 表示用于数据存储的一种有效的数据结构，Hash 表等查找复杂依赖于 Hash 值算法的有效性，在最好的情况下，hash 表查找复杂为?
34. 以下哪个不属于单向哈希表的特征 () B
它把任意长度的信息转换成固定的长度输出
它把固定的信息转换成任意长度信息输出
根据特定的哈希值，它可以找到对应的原信息值
不同的信息很难产生一样的哈希值
35. 假定有 k 个关键字互为同义词,若用线性探测法把这 k 个关键字存入哈希表中,至少要进行多少次探测?()D
k-1 次
k 次
k+1 次
 $k(k-1)/2$ 次
36. 在哈希查找中,"比较"操作一般也是不可避免的() 错
37. 以下属于逻辑结构的是()C
顺序表
哈希表
有序表
单链表
38. 现有一完全的 P2P 共享协议，每次两个节点通讯后都能获取对方已经获取的全部信息，现在使得系统中每个节点都知道所有节点的文件信息，共 17 个节点，假设只能通过多次两个对等节点之间通讯的方式，则最少需要 () 次通讯 30
39. 采用哈希表组织 100 万条记录，以支持字段 A 快速查找，则 ()
理论上可以在常数时间内找到特定记录
所有记录必须存在内存中
拉链式哈希曼最坏查找时间复杂度是 $O(n)$
哈希函数的选择跟 A 无关
40. 采用开放定址法处理散列表的冲突时，其平均查找长度？ A
高于链接法处理冲突
高于二分查找
低于链接法处理冲突
低于二分查找

41. 下述有关 hash 冲突时候的解决方法的说法，错误的有？ B
通常有两类方法处理冲突：开放定址(Open Addressing)法和拉链(Chaining)法。
开放定址更适合于造表前无法确定表长的情况
在用拉链法构造的散列表中，删除结点的操作易于实现
拉链法的缺点是：指针需要额外的空间，故当结点规模较小时，开放定址法较为节省空间
42. 已知一个线性表 (38, 25, 74, 63, 52, 48)，假定采用散列函数 $h(\text{key}) = \text{key} \% 7$ 计算散列地址，并散列存储在散列表 A【0...6】中，若采用线性探测方法解决冲突，则在该散列表上进行等概率成功查找的平均查找长度为 2
43. 对于线性表 (7, 34, 55, 25, 64, 46, 20, 10) 进行散列存储时，若使用 $H(K) = K \% 9$ 作为散列函数，则散列地址为 1 的元素有 () 个 4, [7, 7, 1, 7, 1, 1, 2, 1] 没有告诉处理办法
44. 下面属于构造散列函数的方法是 () ABCD
直接定址法
数字分析法
乘余取整法
平方取中法
45. 解决 hash 冲突的方法描述错误的有？ D
A 开放定址法解决冲突的做法是：当冲突发生时，使用某种探查(亦称探测)技术在散列表中形成一个探查(测)序列。沿此序列逐个单元地查找，直到找到给定的关键字，或者碰到一个开放的地址(即该地址单元为空)为止。
B 拉链法解决冲突的做法是：将所有关键字为同义词的结点链接在同一个单链表中
C 拉链法处理冲突简单，且无堆积现象，即非同义词决不会发生冲突，因此平均查找长度较短
D 当结点规模较大时，开放定址法较为节省空间
46. 下列选项哪些是正确的 D， effective C++ “条款 08：别让异常逃离析构函数”
对于 unordered_map 和 map 这两个容器，迭代器的有效性皆不受删除操作影响
对于 unordered_map 和 map 这两个容器，迭代器的有效性皆不受插入操作影响
为了保证代码的异常安全性，应该避免在构造函数中抛异常
为了保证代码的异常安全性，应该避免在析构函数中抛异常
47. 在哈希查找中,"比较"操作一般也是不可避免的()对
48. Assume you have an object to describe customer data: { ID (7 digit numeric) Family Name (string) Account Balance (currency) } If you have 500,000 Chinese customers records represented by instances of this object type , what set of data structures is best to get fast retrieval of customers (1) get IDs from Name and (2) get Name from ID? B
(1) Tree with Hash(100 bucket) at leaves (2) Tree with linked list at leaves.
(1) Tree with linked list at leaves (2) Array.
(1) Tree with linked list at leaves (2) Hash(10,000 buckets)
(1) Sort linked list (2) Array.
49. 随着装填因子 a 的增大,用闭哈希法解决冲突,其平均搜索长度比用开哈希法解决冲突时的平均搜索长度增长得慢()错
50. 下面哪个不是用来解决哈希表冲突的开放地址法？ C (拉链探测法错误，只拉链不探测)
线性探测法 线性补偿探测法 拉链探测法 随机探测法

51. 下列哪两个数据结构，同时具有较高的查找和删除性能？（）CD
 有序数组
 有序链表
 AVL 树
 Hash 表
52. 既希望较快的查找又便于线性表动态变化的查找方法是()C
 顺序查找
 折半查找
 索引顺序查找
 哈希法查找
53. 执行()操作时,需要使用队列做辅助存储空间 B 深度优先搜索要借助栈, 广度优先搜索要借助队列
 查找哈希(Hash)表
 广度优先搜索网
 前序(根)遍历二叉树
 深度优先搜索网
54. 已知某个哈希表的 n 个关键字具有相同的哈希值, 如果使用二次探测再散列法将这 n 个关键字存入哈希表, 至少要进行____次探测。E
 $n-1$
 n
 $n+1$
 $n(n+1)$
 $n(n+1)/2$
 $1+n(n+1)/2$
55. 下列关于线性表, 二叉平衡树, 哈希表存储数据的优劣描述错误的是? D
 A 哈希表是一个在时间和空间上做出权衡的经典例子。如果没有内存限制, 那么可以直接将键作为数组的索引。那么所有的查找时间复杂度为 $O(1)$;
 B 线性表实现相对比较简单
 C 平衡二叉树的各项操作的时间复杂度为 $O(\log n)$
 D 平衡二叉树的插入节点比较快
56. 有一个数组 (53,83,18,59,38,35), 依次将其存储在 hash 表中, 其中哈希函数为 $h(k)=k\%7$, 如采用线性探测 (每次向后查找 1 位) 的方式解决冲突, 则该 hash 表上查找 38,35,53 访问 hash 表的表项次数分别为 5, 2, 1。

数据结构	查找	插入	删除	遍历
数组	$O(N)$	$O(1)$	$O(N)$	—
有序数组	$O(\log N)$	$O(N)$	$O(N)$	$O(N)$
链表	$O(N)$	$O(1)$	$O(N)$	—
有序链表	$O(N)$	$O(N)$	$O(N)$	$O(N)$
二叉树 (一般情况)	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(N)$
二叉树 (最坏情况)	$O(N)$	$O(N)$	$O(N)$	$O(N)$
平衡树 (一般情况和最坏情况)	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(N)$
哈希表	$O(1)$	$O(1)$	$O(1)$	—