作者：星\_\_尘  
链接：<https://www.nowcoder.com/discuss/425581?type=post&order=time&pos=&page=3&channel=-2&source_id=search_post>  
来源：牛客网  
  
题目全部来自牛客网字节跳动面经，楼主整理了2020年的三十余篇面经，在多次问到的题目后标注了次数，限于篇幅，答案没有上传，如果牛友们需要希望点个赞呀😁😁，楼主整理后分次上传答案。

1. **算法题：买卖股票的最佳时机（只能有一次买卖，可以最多两次买卖，不限次数）（4）**
2. **算法题(leetcode55题)：给一个数组，例如[1,2,3,4,5]，a[i]表示在该位置可以向前行走的最大距离，判断是否可以到达数组的最后一个元素。**
3. **场景题：让你设计一个微信发红包的api，你会怎么设计，不能有人领到的红包里面没钱，红包数值精确到分。**
4. **AB两个排序数组，原地合并数组。（A当中穿插一些无效数字怎么处理？）**
5. **剑指原题，剪绳子。**
6. **排序数组，平方后，数组当中有多少不同的数字（相同算一个）。**
7. **一个数据先递增再递减，找出数组不重复的个数，比如 [1, 3, 9, 1]，结果为3，不能使用额外空间，复杂度o(n)**
8. **高考成绩2000万数据，分数0-750，如何快速知道你的排名，如何知道任一分数排名 --->桶排序 （3）**
9. **两根香，一根烧完1小时，如何测量15分钟---->开始时一根香两头点着，一根香只点一头，两头点着的香烧完说明过去了半小时，这时将只点了一头的香另一头也点着，从这时开始到烧完就是15分钟。**
10. **两个链表，可能相交，找出相交的节点，给出证明（2)**
11. **写一个函数，求平方根，函数参数为目标数字和精度，测试案例 fn(4.1,0.001) fn(501.1,0.001) fn(0.045,0.001)**
12. **场景题：需求：谁关注了我，我关注了谁，谁与我互相关注。表该如何设计，索引怎么建。查询语句怎么写**
13. **10亿个数字，取最小的100个数**
14. **1亿个正整数,范围是0-42亿。求出现次数是2的数字，空间复杂度**
15. **剑指offer 从上往下打印二叉树（层序遍历）（2)**
16. **蛇形遍历二叉树(2)**
17. **leetcode 链表求和**
18. **给定一个 0-4随机数生成器 如何生成0-6随机数**
19. **写代码： 二叉树的最近公共祖先 leetcode 236 稍有不同，原题的2个节点，面试是多个节点，算法的时间复杂度 (2)**
20. **写代码： 二叉树中的最大路径和 leetcode 124**
21. **算法：快排（3)**
22. **算法：二叉树的前序遍历非递归**
23. **算法：二叉树的后序遍历非递归(2)**
24. **求数组的最长连续递增数列，如：4， 200， 3， 1， 100， 2。结果是1 2 3 4，也就是说顺序可以打乱。（leetcode 128）**
25. **智力题，海盗分金币。**
26. **算法：接雨水（leetcode 42）**
27. **算法：有一个IP地址库，假设有几十万条ip，如何判断某个ip地址是否在这个库中？**
28. **编程题：求二叉树根节点到叶子结点的路径和的最小值（leetcode）**
29. **反转链表 ---->反转链表升级版（每k个反转一下）（4）**
30. **场景题：2g内存，要求一个10g文件的中位数**
31. **LeetCode 59题 螺旋矩阵II（同类：螺旋打印矩阵）**
32. 数据结构：（讲解你了解的数据结构）提到heap，让手写heap
33. **算法：中文数字转阿拉伯数字，字符串处理问题**
34. **算法：中文数字转阿拉伯数字，字符串处理问题**
35. **重建二叉树（剑指offer第7题）**
36. **路径总和 leetcode 112及其延伸**
37. **（LeetCode113题）给定一个二叉树和一个目标和，找到所有从根节点到叶子节点路径总和等于给定目标和的路径。**
38. **单例模式，手写双重检验单例模式 懒汉式，DCL（8）**
39. 合并区间 （LeetCode56题）
40. **翻转字符串中的单词 leetcode151原题 并且只能用O(1) extra space**
41. **和为s的连续正整数序列（剑指offer57-II)**
42. **介绍下二分查找，如果我想在链表中使用二分查找，怎么做比较好？-->跳表**
43. **介绍下归并排序，时间复杂度多少**
44. **LRU算法知道吗，怎么实现的?**
45. **数据库连接池怎么设计?**
46. **代码题,版本数字比较，比如"1.10.0"版本比"1.8.1"版本新，不允许使用split等函数**
47. **某一个大文件被拆成了N个小文件，每个小文件编号从0至N-1，相应大小分别记为S(i)。给定磁盘空间为C，试实现一个函数从N个文件中连续选出若干个文件拷贝到磁盘中，使得磁盘剩余空间最小。**
48. **场景题：redis设置高并发抢单一的东西，如何避免高并发对一个键进行访问**
49. **编程：最大栈**
50. **场景题，分布式多个机器生成id，如何保证不重复?**
51. B+树和红黑树， 红黑树和一般的平衡二叉树，增、删、改、查的过程和效率、时间复杂度
52. **递增数组，找出和为k的数对**
53. **二叉树各层节点数，递归、非递归，时间、空间复杂度**
54. **输出给定数字下一个比它大的数字，比如输入：1234， 输出 1243。**
55. **算法：输入List<String>，删除当中形如”1\_”的，返回原来的List (2)**
56. **一个无序数组，从小到大找到第一个缺的数，比如[8 2 4 3 6 9 7 11 12],第一个缺的就是5 （2）**
57. **场景题：一个硬币，正面概率0.7，反面概率0.3，现在有一瓶水，怎么掷能让两个人公平的喝到水(4）---->抛两次，先正后反A喝，先反后正B喝**
58. **两个栈实现一个队列**
59. **二叉树右视图（2）**
60. 概率：54张扑克牌，平均分成3份，大小王在一份的概率
61. **有序有重复数组，给定target确定范围**
62. 3sum （2） 注意时间复杂度
63. 扫码登录是如何实现的？
64. **思考题：64匹马，8个跑道，选跑最快的4匹马需要比赛多少次。**
65. 概率：两个人轮流抛硬币，先抛到正面的赢，问先抛的人赢的概率（2）
66. **三个线程循环打印ABC**
67. **数组中第K大的数（LeetCode215题）**
68. top k，堆的各种时间复杂度
69. 狼和羊的故事
70. 二叉排序树找第k大的元素
71. 链表相加，反转过来怎么加
72. 穿墙术，三维的bfs，类似走迷宫
73. 有一个会议室里有一个录音麦，有n个人抢着说话，麦只能录到声音最大的人，给定每个人开始的说话时间s，结束的说话时间t，说话音量vol，然后求这个麦最后录到的声音序列。 数据范围:n 2e5。 s,t,vol 1e9
74. 场景设计：系统中有几万个任务需要在各自的特定时刻触发执行，怎么做？
75. **三道leetcode原题：236，240，面试题48略微更改**
76. 一个很复杂的场景，大概是微博用户有个关注列表，以及知道各种大V的动态，设计数据结构啥的，到最后在用户这边显示一个更新微博列表。
77. 场景题：游戏里，一个玩家可能吃到很多种debuff，每个debuff有一定持续时间。给你一个定时器，怎么设计数据结构，来维护玩家的debuff状态？
78. 红黑树，B+树，跳表应用场景
79. 代码题：leetcode 688 棋盘算概率问题
80. 情景题，一个5T的文件，里面全是id，1-10^9 ，如何计算不同id的个数
81. 算法题，一个有序数组，从随即一位截断，把前段放在后边，如 4 5 6 7 1 2 3求中位数
82. 算法题，一个形如 123456789101112……的字符串，输入一个n（很大很大），输出字符串第n个字符
83. **算法题，一辆公交车，有m站，最多坐n人，输入一路上票的信息（即上车下车站），输出会不会超载**
84. 算法题，全排列
85. 坐标系中有一个球桌，四个角坐标：(0,0), (0,4), (2,4), (2,0)，一颗球在(1,1)，请问从哪些角度可以射入洞内（可无限次碰撞）？
86. **求完全二叉树的节点个数，小于O(n)，并分析复杂度**
87. **链表实现一个栈**
88. **（1） Leetcode 134 （2）Leetcode 860。**
89. **算法：手写jdk中的优先级队列 PriorityQueue（最小堆）**
90. **剑指offer62:圆圈剩下的数字（约瑟夫环问题）**
91. **给出一个数组nums，一个值k，找出数组中的两个下标 i，j 使得 nums[i] + nums[j] = k.**
92. 说说你知道的设计模式，说说项目里用到的设计模式，说说策略模式，设计一个下棋的场景问如何结合设计模式使用，设计模式什么时候继承，什么时候委托？
93. **以下代码题输出什么？（巨坑，输出100，从泛型+向上转型+map+equals原理上想）**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | Map<Short, String> map = new HashMap<>();  for(short i = 0; i <100; i++) {       map.put(i, String.valueOf(i));       map.remove(i-1);  }  System.out.println(map.size()); |