

1. Why Amazon?

2. Most challenging project.

3. Reverse Linked List

4. Differences between vector and linkedlist (synchronization)

5. valid binary search tree

6. differences between a graph and tree

coding 和 knowledge 部分都问了一下时间空间复杂度的问题, valid binary search tree 部分简单的 follow up 了一下有没有别的做法

7. 第一道是 leetcode 原题, validate parentheses () {} [] 但这个 string 中间有各种字母, 需要排除掉它。

8. 第二个是 design problem 不用写代码, 给 car Honda, color, engine, Vehicle(单车也包括在内) drive. 让写他们之间的关系。

写几句话就可以了。

9. 然后是 Design!!! 题目是 BlackJack Game()... 妥妥的跪嘛。。要求花一点时间思考, 然后拿出了准备好的 Design 题, 对着答案也没抄好, 解释的更是烂。。

10. 先是聊数据结构, LinkedList、ArrayList、BST、HashMap, 各种操作的复杂度, 比较。然后具体聊了 HashMap 的实现。

11. Code problem 是 Two Sum, 不过说到重复的输出 (类似如果有 {1, 2, 2, 3} 求和等于 5, 有多个 duplicates 满足需要全返回 { (2, 3), (2, 3) }, 我说意义何在, 他说就是要. two sum, 找所有的 pair, 所有可能的 case

12. 第二个是从所有的 Http file 中找手机号码, 考察 grep 和 regular expression

13. 第三个是 pocket cards design, 想到什么说什么~

14. given an array, find the 2 largest number.

15. questions: top k numbers, Priority queue, how to implement, insert get time complexity?, O(k) space? -- queue of length k -- how to add elements?

16. hashmap, how to store in memory, time complexity

17. implement class TreeIterator() : hasNext(); next(); 还有 constructor

18. //Without using the "/" operator, create a function/method in a strongly-typed language of your choice that takes two integers and returns their quotient, exactly mimicking the behavior of the / operator. Don't worry about creating a main function, class or imports. Assume you have access to any library functions you might need -- unless there's a library function that performs this already.

19. 很简单, 两个数组, 然后找出两个数组中都存在的数。

20. 太点背，leetcode 上都没有。。。。考的是 $m \times n$ 二维数组，找一个指定数字是否存在然后返回 boolean

21. 给你一个 Integer 的 Stream，在 Stream 结束的时候随机等概率返回其中的一个元素。

22. implement set of stacks

23. head of linkedlist cycle

24. 然后问了 ArrayList 和 linkedlist

25. 找出链表倒数第 3 个元素。当时脑子一抽，就写了一个遍历两边的方法，time 复杂度是 $O(2n)$ ，然后他说要 $O(N)$ ，磨蹭了一会用两个指针，写了一个 $O(n)$ ，后来还要优化，就说三个指针？然后就下一题了

26. 他给了一个 api，大概是用来读入两个 file 的，然后找到 sharelist，刚开始没读懂意思，还问他怎么读 file，他说就用这个 class，然后写完用了 hashset 他还说为啥用 hashset，就说删除 duplicates，他还奇怪，后来好像是我理解错了，后来时间过了随便问了问题就接了。感觉自己特别捉急。。。

27. HashSet

28. HashMap 里面很多元素出现同样的 Hash code 怎么办，然后我解释了 linear chaining, linear probing, quadratic probing.

28. 如果往 HashSet 里面添加重复的元素会发生什么。我说不会增加元素数量，HashSet 里面元素都是独一无二的。

29. 写代码，third largest. 我按照 Second largest 的方法写的。设定三个变量，first, second, thrid

30. Leetcode 原题 Triangle. 不过这里是返回最大数。由于已经做过了，开始装逼，说像是个最短路径问题，另外建立一个三角形，从上往下，然后把所有元素都过一遍。让我开始写代码。然后我说还能 inplace. 写了几行，

31. longest substring repeating characters

32. Abstract with interface

33. polymorphism

34. 二叉树的平衡判断

35. two sum of 10, two sum of target, 返回 true/false

follow up: 返回下标

36. app to track busses and busstops in a city

大概就设计了一下类时间就没有了（前面项目经验扯淡花了很多时间，本人听力简直渣，一直在确认。。。）

Bus: Coordinate

BusStops: Coordinate, City

City: list<BusStops>

Coordinate: int x,y

37. hashtable和hashmap的区别，这难道不是 java 的东西么？我是用 C++的啊，直接回答不知道。然后他就说"it's ok, it's ok"....

38. 一个 array，找到一个 index 把这个 array 平均的分成两部分

39. arraylist和linkedlist的区别，hashmap 和 treemap 的区别，dfs 和 bfs 的区别，然后就是算法题：

40. 定义每个节点的 sum 为以这个节点为根的所有节点上的数字和。找到一个节点，它的 sum 离 root 的 sum 的一半最近。我写了

一个遍历两次二叉树的解法，先遍历一遍求每个节点 sum，然后再遍历一遍找目标节点。

问了时间复杂度，然后问怎么改进。提示说 sum 是

逐层递减的。面试的时候有点紧张，思路不太清楚，就说了先判断应该选择哪个孩子，然后从这个孩子开始往下遍历。然后他说 you get the

idea，代码没写完就让问问题了。其实应该再判断一下选择这个孩子之后 $\text{abs}(\text{child.sum} - \text{half})$ 与 $\text{abs}(\text{parent.sum} - \text{half})$ 的关系。

41. 如何从链表中 remove 一个 node (没让 coding，直接说想法，其实很简单 $O(n)$ 两个指针)

42. 其实就是给一个 sorted array 0 0 0 1 1 1 1 然后找出第一个为 1 的数

然后由于被印度哥们折磨的要死，所以瞬间紧张了，不过还是淡定的说出了，binary search 的解法（竟然悲剧的把 $\text{start} \geq \text{end}$ ）擦，我也是醉了。估计要跪就跪在这里了.....

写完后，印度哥们给出了三个 case，测试，结果是过了。之后讨论 cornercase，我说第一个情况是 array 为空，要返回 null 或者抛出 exception，然后就是比如数组全 0 的时候，我说可以让 $\text{index} = \text{MAX_VALUE}$ ，如果 BS 之后还是这个值就抛出异常（此时印度哥们终于说了句 awesome）.....

43. Maximum Subarray.

44. reverse 一个 string，写一个函数，不难

45. hashmap的insert 和get 各自是什么复杂度，这个也不难

46. 单向 linkedlist 查环，用了 fast, slow pointer 做法

47. 有个数组长度是 n，里面的元素是 from 1 to n + 1，貌似是一个元素少了，找出 missing

元素，一开始我以为是找出 index，所以说 swap 之类的，然后他说要返回的是 value，我说用 bucket sort，然后他说有个更好的，把 array traverse 一遍加起来，然后减一下就是了。

这道题没让 coding，他说，I just came up, never mind, 是第二个题的 follow up，说还有一种 $O(n)$ 的做法，我说是用 hashmap 做，然后他说正确，就别 coding 了。

48. Fibonacci return 小于 M 的最后 N 个数，然后问了一下优化，

49. corner case: 长度小于 2 怎么办，我一开始返回 Integer.MIN_VALUE，他说不太好，让 improve，没想出来。。他说可以 throw exception T_T

50. 现在有 thousand of files, 要找出含有 credit card number 的 file, 如何找

51. find highest order digit of a byte, 判断一个字节的最高位是什么。他说可以 pseudo code, 我直接给了 code。然后问 what is then endian here. just curious (一开始不知道他问什么，不知道 endian 这个词。原来是哪边大哪边小的问题。。他说 it's ok, your code works.)

55. follow up 第一题，如果第一题要找 3th largest or kth largest, 如何做? (min heap, 问要不要写 code, 他说随意，解释到就行，就边解释边写了点 pseudo code, 他说 cool, 时间差不多，问我问题吧)

56. 问我一系列关于亚马逊的问题: Why Amazon? 你对我们的老大了解多少? (我只知道他叫 jeff) 你和亚马逊不得不说的故事? (妈蛋除了买过书，我还能和你们有什么 story)

57. Process 和 thread 的区别

58. Lock, dead lock. What will happen if there are too many locks in a program...(这个不知该怎么答)

59. Design pattern (singleton, factory)

60. Encapsulation and polymorphism

61. find the number that occurs odd number of times

62. delete a node in a circular singly linked list

63. find nth fibonacci number

64. 树的 Zigzag Level Order。

65. 斐波那契数列。

直接用三个变量解决。后来想想，其实应该先用 DP 那个方法解决，然后再用三个变量的方法解决，这样能磨蹭一下时间，能多码一些，也展现一下思考过程。

然后优化边界条件，然后各种优化，问如果数特别大如何解决，各种说错，然后在他再三提示下才说出来。

66. 一个 array 数字乱七八糟，但是每个数都出现两次，只有一个一次，找出幸运儿！

67. Implement a map and its functions insert(), find()

68. Find the middle of a singly linked list.

69. 问平时用什么数据结构，我按当年数据结构课讲义扯了几个。然后问 `vector` 和 `linked list` 有什么区别。

70. 编程题，给一个整数数组，返回最大值。然后又让改成返回第二大、第三大的值。我直接用 `heap` 写了个 `topK` 然后调用了一下，他说不用这么麻烦，就让我用数组操作写，那只好老老实实的写喽。

71. 设计题，问 `interface` 是什么意思，然后设计一个 `Deck` 类。

72. `swap nodes in a linked list in pairs`

73. Given Tree and Node `n` and `int k`, print all node which are at physical distance $\leq k$ from `n`.

74. `get highest water level`, 这题还没刷到，动态规划悲剧了。。。

75. 给一个 `int array` 打印 `bar height`

76. 然后是leetcode的`SameTree`，判断两个树是不是相同的
分别用递归和迭代实现出来

77. 问的是 `leetcode` 原题，找两个 `linked list` 从哪个 `node` 开始重合。但是我真心渣

78. 然后问了一道 `sudoku` 的题，就是给了一个已经填满的 `9*9` 的 `sudoku`, 让我判断这个 `sudoku` 是否 `valid`。

follow up, 一个合法的 `sudoku` 的边长有什么特点？（`n` 必须是某个整数的平方）

follow up, 把程序改成适合任意合法长度的 `sudoku`。（也就是说不一定要填 1-9，也可以是比如 1-16, 1-64 什么的，随长度定）

follow up, 把程序改成也可以判断有 `cell` 没填的 `sudoku` 是否 `valid`。（只要已填的部分未重复就好）

然后问了 `hashmap` 和 `bst` 的基础知识。

79. `Min in rotated sorted array`

80. 1. `Shuffle card`

2. `Lowest common Ancestor`

1. `Min in rotated sorted array`

2. `Merge K sorted array`

81. `Array Length encoding`: 给定 `binary` 数组(比如[1010]), 计算每个 `digit` 数量, 返回这种形式([11011101]).

82. 第一题是跟 `duplicates` 有关的, 给你一个很大很大的文件, 文件每行是一个 `integer`, 如何最快地找出文件中有没有 `duplicate`." 我说用个 `hash table`, 但面试官又说不能编程。最后跟面试官说不编程不会做了。。。面试官说要用 `commands`。然后跟我说要用

cat,sort,unique,diff 这几个。

83. 这道题还跟 duplicates 有关，写一个 method，parameters 是一个 array 还有它的长度，返回有没有 duplicates。给的 method 上面居然写着 $0 < arr[i] \leq \text{length}$ for all $0 \leq i < \text{length}$ 。然后我就写了 direct-address 的方法。“空间是多少”。我说 $O(n)$ 。“可不可以少用点内存”。我说用 bitset。“怎么 set 还有 test 一个 bit”。我就给他讲了 bit manipulation。“现在空间多少”。我说 $n/32$ 。面试官说答得不错，表示很满意。然后又问可不可以不用额外空间。在面试官指导下想了会也没答出来。然后他跟我了解法，真是醉了。。。

84. count string 中的 words

85. bfs tree, print in levels

86. tree serialization and de-serialization

87. // Write a function to “flood fill” a bitmap, function should allow controlling level of flood fill.

// A[5][5] of integers where each cell can have only 0/1. If it's 0, its not flooded yet, if it's 1, its already flooded.

// water at point p at time t1,

// p, p's neighbors at time t2 (t1+x)

// p, p's neighbors, p's neighbors' neighbors -> at time t3 (t2+x)

// current cell (i, j) => i-1, j or i, j+1, or i, j-1, or i+1, j or i+1, j+1 or i-1, j-1

// input: 2d array(filled, but), starting cell (i, j), level of flooding (1,2,3....)

88. 第一个问题是在 linux 里面一大堆文件，要找文件里的东西——我说用 find 命令找文件，cat 打开文件，然后用 regex 匹配。

89. 第二题是相同数组里找相同 ID (duplicate 问题)。

90. compare version number

91. valid palindrome

92. nth fibonacci (recursion, dp)

93. input 是一个 string list, 判断他里面的元素是不是 anagram, 如果都是 anagram 返回 true, 其他返回 false, 大小写有区别。写完给他解释一遍 code, 然后他问我有没有 bug, 我看了半天胆怯的说没有, 他也没跟我继续讨论 corner case 什么的, 让我分析一下时间复杂度。

94. 然后就是数据结构的问题: arraylist 和 linkedlist 的区别, hashmap 和 treemap 的区别, dfs 和 bfs 的区别, 然后就是算法题:

给一棵二叉树，定义每个节点的 **sum** 为以这个节点为根的所有节点上的数字和。找到一个节点，它的 **sum** 离 root 的 **sum** 的一半最近。