https://instant.1point3acres.com/thread/302577
2. SQL 部分
adv_info:
advertiser_id
ad_id
spend:The Advertiser pay for this ad

ad_info:
ad_id
user_id
price: The user spend through this ad (Assume all prices in this column >0)

Q1: The fraction of advertiser has at least 1 conversion?
Q2. What metrics would you show to advertisers (其实就是在问 ROI), 用 SQL 实现

```
select count(distinct advertiser_id)/(select count(distinct advertiser_id from adv_info)
from adv_info adv
left join ad_info ad
on adv.ad_id = ad.ad_id
where ad.price > 0

--ROI

select adv.advertiser_id,
       cost = sum(spend),
       rev = sum(isnull(ad.price,0)),
       ROI = sum(spend)/sum(isnull(ad.price,0))
from adv_info adv
left join ad_info ad
on adv.ad_id = ad.ad_id
group by adv.advertiser_id
```

https://instant.1point3acres.com/thread/297747
发一下两轮电面的面经。。求攒人品。。。。

第一轮
中国大叔很 nice。。
SQL
Question: Write a query to list the departments that have a total combined salary greater than $40,000.
Question: Find the highest paid employee in the company.
Question ： Nth highest paid

Coding
a = (1, 0, 0, 2, 0, 0, 5)
b = (0, 0, 1, 3, 0, 1, 4)
dot product = 1*0+ 0*0 +... + 5*4 = 26

第二轮：

印度大叔，没有口音。。人很 nice。

lc125

lc79

```
class Solution:
    def exist(self, board, word):
        """
        :type board: List[List[str]]
        :type word: str
        :rtype: bool
        """
        if(len(word)==0):
```

```
            return True
        if(len(board) == 0):
            return False
        elif(len(board[0]) == 0):
            return False

        used = [[False]*len(board[0]) for _ in range(len(board))]

        for i in range(len(board)):
            for j in range(len(board[0])):
                if (self.findnext(board, i, j, word, 0, used)):
                    return True
        return False

# i, j current char to compare
def findnext(self, board, i, j, word, kword, used):
        if(kword == len(word)):
            return True

        if(i<0 or i>=len(board) or j<0 or j>=len(board[0])):
            return False
        if(used[i][j]):
            return False

        if(board[i][j] == word[kword]):
            used[i][j] = True
            kword += 1
            if (self.findnext(board, i-1, j, word, kword, used) or
                    self.findnext(board, i+1, j, word, kword, used) or
                    self.findnext(board, i, j-1, word, kword, used) or
                    self.findnext(board, i, j+1, word, kword, used)):
                return True

        used[i][j] = False
        return False
```

https://instant.1point3acres.com/thread/302618
facebook 师兄内推，可能是写了比较详细的 research proposal，hr 发了邮件，只有两轮面试，第一轮面 coding,第二轮 team match。。。

第一轮 coding 是一个三哥面的，两道题：
第一个是求一个 matrix 中任意一个 sub matrix 中 element 的和，lc 308，不过需要用积分图(integral image)做，也就是 read heavy,要求查询复杂度是 O(1),问了 build 积分图的复杂度(O(hw)),写代码，要求 bug free
第二个是问有 n 个 pair (x1, y1), (x2, y2) .. (xn, yn),给定一个 target k，要求返回任意两对 pair (xi, yi), (xj, yj),使得 xi+xj = yi+yj = k.
一开始我用了 sort+2 pointer 做，复杂度 O(nlogn),三哥表示能 work，让我写了伪代码，接着 follow up，问能不能 O(n),我说可以，用一个 dict 存 xi(我用 python) 对应每个 xi 存的 value 是一个 set,放 yi，一遍循环，如果 k-xi 在这个 dict 并且 k-yi 在这个 xi 的 value set 里，就返回，不知道是不是 lc 原题，大家可以在下面给出题号
第一轮面试遇到三哥还是挺犯怵的，而且我写的第一遍并不是 bug free,写完他立刻说，你的代码有 bug,说了三遍，不过后来我都自己查了出来并改正

第二轮是面 researh，类似聊天，没有什么技术含量。。

第二天，也就是今天收到了面试通过的电话，感谢三哥放我一马

时间线:
12.19/2017: 网投简历
12.20/2017: 收到了面试通知和 schedule
1.4/2018: 下午连续面试两轮
1.5/2018: offer

```
a = [[1,1],[1,2]]

b = {}

for item in a:
    k = item[0]
    v = item[1]
    if (b.get(k) is None):
        b[k] = [v]
    else:
        b[k].append(v)
```

https://instant.1point3acres.com/thread/298407
Longest Substring Without Repeating Characters 24.5% Medium
4
Median of Two Sorted Arrays 22.4% Hard
8
String to Integer (atoi) 13.9% Medium
14
Longest Common Prefix 31.6% Easy
18
4Sum 27.2% Medium
21
Merge Two Sorted Lists 39.4% Easy
29
Divide Two Integers 15.9% Medium
31
Next Permutation 28.9% Medium
34
Search for a Range 31.5% Medium
36
Valid Sudoku 36.6% Medium
42
Trapping Rain Water 37.2% Hard
46
Permutations 45.5% Medium
47
Permutations II 33.9% Medium
55
Jump Game 29.5% Medium
62
Unique Paths 42.0% Medium
65
Valid Number 12.9% Hard
81
Search in Rotated Sorted Array II 32.7% Medium
84
Largest Rectangle in Histogram 27.2% Hard
100
Same Tree 47.1% Easy
101
Symmetric Tree 39.6% Easy
105
Construct Binary Tree from Preorder and Inorder Traversal 32.9% Medium
108
Convert Sorted Array to Binary Search Tree 42.9% Easy

114
Flatten Binary Tree to Linked List 35.8% Medium
115
Distinct Subsequences 31.9% Hard
124
Binary Tree Maximum Path Sum 26.7% Hard
126
Word Ladder II 14.5% Hard
138
Copy List with Random Pointer 26.0% Medium
151
Reverse Words in a String 15.7% Medium
152
Maximum Product Subarray 26.3% Medium
167
Two Sum II - Input array is sorted 47.2% Easy
189
Rotate Array 25.0% Easy
217
Contains Duplicate 46.4% Easy
227
Basic Calculator II 29.6% Medium
241
Different Ways to Add Parentheses 45.1% Medium
270
Closest Binary Search Tree Value 40.1% Easy
304
Range Sum Query 2D - Immutable 26.3% Medium
305
Number of Islands II 39.5% Hard
317
Shortest Distance from All Buildings 34.4% Hard
322
Coin Change 26.7% Medium
333
Largest BST Subtree 30.7% Medium
336
Palindrome Pairs 26.7% Hard
348
Design Tic-Tac-Toe 45.8% Medium
349
Intersection of Two Arrays 47.6% Easy
350
Intersection of Two Arrays II 44.8% Easy

443
String Compression 37.1% Easy
449
Serialize and Deserialize BST 42.4% Medium
454
4Sum II 47.4% Medium
499
The Maze III 32.9% Hard
518
Coin Change 2 33.7% Medium
546
Remove Boxes 34.5% Hard
560
Subarray Sum Equals K 40.6% Medium
567
Permutation in String 36.6% Medium
640
Solve the Equation 38.9% Medium
652
Find Duplicate Subtrees 36.1% Medium
662
Maximum Width of Binary Tree
补充一道 318. Maximum Product of Word Lengths
360. Sort Transformed Array