https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=490785&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Pass** | fresh grad 应届毕业生

第一题是一个给了 list of string 让排序， 要重写 comparator 然后 sort in lexical order， 看题看了十分钟。。。

第二题是给了一个长 string 让分割成互不重叠的 substring， 我的做法是， 先 create interval， 再 merge interval， 感觉之间应该做过类似的。

以下是我总结了地理的面经和链接，有一些是之前小伙伴提到过的前十条，有一些是我新加的。

1. 卡车装 M 个箱子， N 个地点 List<Integer> M<N
列出最近的 M 个位置。 要注意输入不正常的情况，比如只有一个输入
http://www.1point3acres.com/bbs/thread-289277-1-1.html

大意是说 Amazon 开了新的 warehouse，要用卡车给其他地方送货

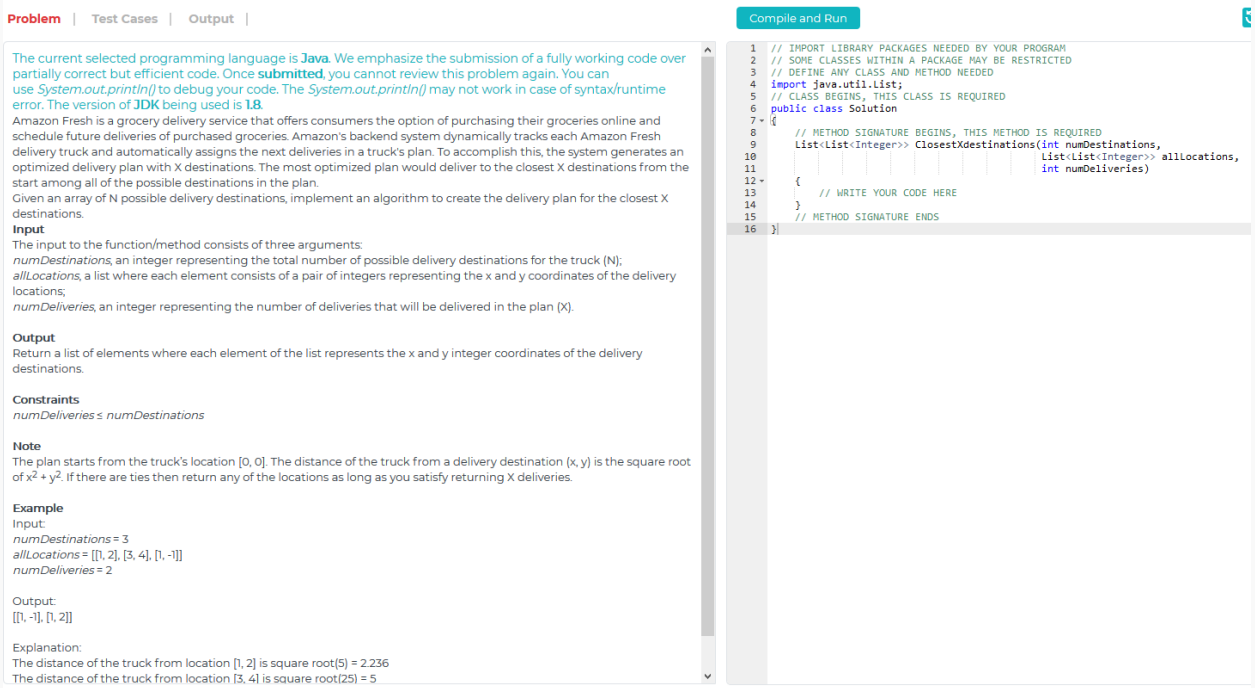
参数:
int N, 代表总共有 N 个地点
List<List<Integer>> 地点的坐标
int M,代表需要送的 crate 数量

output：一个 List<List<Integer>> 代表送货的地点坐标 x,y
其实就是让你计算距离卡车最近的 M 个地点.

需要注意点是题目里面没有给卡车的位置，根据给的例子猜出是原点（0，0）
例 1：N = 3, M = 2, List<List<Ingeter>> 是 [[2,3][3,4],[1,-3]]
output: [[2,3],[1,-3]]

例 2： N=3， M=6， List<List<Integer>> 是[[1,8],[2,4],[8,9],[5,3],[2,7],[3,5]]
output: [[2,4],[5,3],[3,5]]

https://www.1point3acres.com/bbs ... 6orderby%3Ddateline



Problem | Test Cases | Output |

Compile and Run

The current selected programming language is **Java**. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use *System.out.println()* to debug your code. The *System.out.println()* may not work in case of syntax/runtime error. The version of **JDK** being used is **1.8**.

Amazon Fresh is a grocery delivery service that offers consumers the option of purchasing their groceries online and schedule future deliveries of purchased groceries. Amazon's backend system dynamically tracks each Amazon Fresh delivery truck and automatically assigns the next deliveries in a truck's plan. To accomplish this, the system generates an optimized delivery plan with X destinations. The most optimized plan would deliver to the closest X destinations from the start among all of the possible destinations in the plan.

Given an array of N possible delivery destinations, implement an algorithm to create the delivery plan for the closest X destinations.

Input

The input to the function/method consists of three arguments:

numDestinations, an integer representing the total number of possible delivery destinations for the truck (N);

allLocations, a list where each element consists of a pair of integers representing the x and y coordinates of the delivery locations;

numDeliveries, an integer representing the number of deliveries that will be delivered in the plan (X).

Output

Return a list of elements where each element of the list represents the x and y integer coordinates of the delivery destinations.

Constraints

numDeliveries ≤ *numDestinations*

Note

The plan starts from the truck's location [0, 0]. The distance of the truck from a delivery destination (x, y) is the square root of $x^2 + y^2$. If there are ties then return any of the locations as long as you satisfy returning X deliveries.

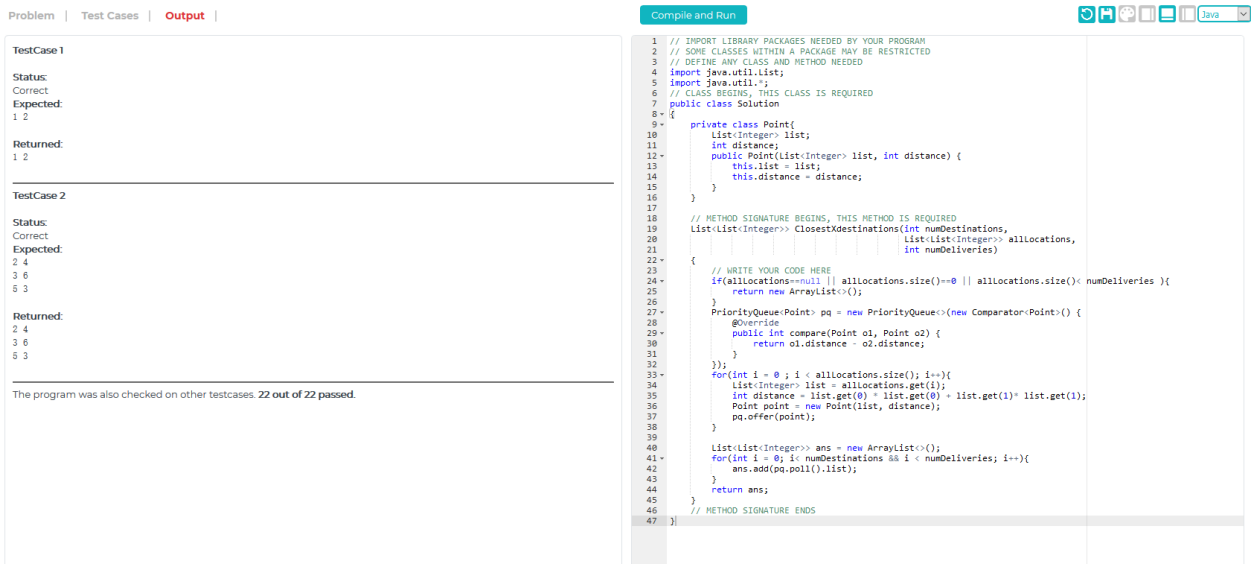
Example

Input:
numDestinations = 3
allLocations = [[1, 2], [3, 4], [1, -1]]
numDeliveries = 2

Output:
[[1, -1], [1, 2]]

Explanation:
The distance of the truck from location [1, 2] is square root(5) = 2.236
The distance of the truck from location [3, 4] is square root(25) = 5

```java
// IMPORT LIBRARY PACKAGES NEEDED BY YOUR PROGRAM
// SOME CLASSES WITHIN A PACKAGE MAY BE RESTRICTED
// DEFINE ANY CLASS AND METHOD NEEDED
import java.util.List;
// CLASS BEGINS, THIS CLASS IS REQUIRED
public class Solution
{
    // METHOD SIGNATURE BEGINS, THIS METHOD IS REQUIRED
    List<List<Integer>> ClosestXdestinations(int numDestinations,
                                             List<List<Integer>> allLocations,
                                             int numDeliveries)
    {
        // WRITE YOUR CODE HERE
    }
    // METHOD SIGNATURE ENDS
}
```



Problem | Test Cases | Output |

Compile and Run

Java

TestCase 1

Status:
Correct
Expected:
1 2

Returned:
1 2

TestCase 2

Status:
Correct
Expected:
2 4
3 6
5 3

Returned:
2 4
3 6
5 3

The program was also checked on other testcases. **22 out of 22 passed.**

```java
// IMPORT LIBRARY PACKAGES NEEDED BY YOUR PROGRAM
// SOME CLASSES WITHIN A PACKAGE MAY BE RESTRICTED
// DEFINE ANY CLASS AND METHOD NEEDED
import java.util.List;
import java.util.*;
// CLASS BEGINS, THIS CLASS IS REQUIRED
public class Solution
{
    private class Point{
        List<Integer> list;
        int distance;
        public Point(List<Integer> list, int distance) {
            this.list = list;
            this.distance = distance;
        }
    }

    // METHOD SIGNATURE BEGINS, THIS METHOD IS REQUIRED
    List<List<Integer>> ClosestXdestinations(int numDestinations,
                                             List<List<Integer>> allLocations,
                                             int numDeliveries)
    {
        // WRITE YOUR CODE HERE
        if(allLocations==null || allLocations.size()==0 || allLocations.size()< numDeliveries ){
            return new ArrayList<>();
        }
        PriorityQueue<Point> pq = new PriorityQueue<>(new Comparator<Point>() {
            @Override
            public int compare(Point o1, Point o2) {
                return o1.distance - o2.distance;
            }
        });
        for(int i = 0 ; i < allLocations.size(); i++){
            List<Integer> list = allLocations.get(i);
            int distance = list.get(0) * list.get(0) + list.get(1)* list.get(1);
            Point point = new Point(list, distance);
            pq.offer(point);
        }

        List<List<Integer>> ans = new ArrayList<>();
        for(int i = 0; i< numDestinations && i < numDeliveries; i++){
            ans.add(pq.poll().list);
        }
        return ans;
    }
    // METHOD SIGNATURE ENDS
}
```

https://1o24bbs.com/t/topic/3152 amazon 全职 Amazon 社招 OA 还是之前的老 2 题

第一题是 Amazon Fresh 送货的

Amazon Fresh is a grocery delivery service that offers consumers the option of purchasing their groceries online and schedule future deliveries of purchased groceries. Amazon's backend system dynamically tracks each Amazon Fresh delivery truck and automatically assigns the next deliveries in a truck's plan. To accomplish this, the system generates an optimized delivery plan with X destinations. The most optimized plan would deliver to the closest X destinations from the start among all of the possible destinations in the plan.

Given an array of N possible delivery destinations, implement an algorithm to create the delivery plan for the closest X destinations.

Input
The input to the function/method consists of three arguments:
numDestinations, an integer representing the total number of possible delivery destinations for the truck (N);
allLocations, a list where each element consists of a pair of integers representing the x and y coordinates of the delivery locations;
numDeliveries, an integer representing the number of deliveries that will be delivered in the plan (X).

Output
Return a list of elements where each element of the list represents the x and y integer coordinates of the delivery destinations.

Constraints
numDeliveries ≤ *numDestinations*

Note
The plan starts from the truck's location [0, 0]. The distance of the truck from a delivery destination (x, y) is the square root of $x^2 + y^2$. If there are ties then return any of the locations as long as you satisfy returning X deliveries.

Example
Input:
numDestinations = 3
allLocations = [[1, 2], [3, 4], [1, -1]]
numDeliveries = 2

Output:
[[1, -1], [1, 2]]

Explanation:
The distance of the truck from location [1, 2] is square root(5) = 2.236
The distance of the truck from location [3, 4] is square root(25) = 5
The distance of the truck from location [1, -1] is square root(2) = 1.414
numDeliveries is 2, hence the output is [1, -1] and [1, 2].

```
TestCase 1

Status:
Correct
Expected:
1 2

Returned:
1 2
```

```
TestCase 2

Status:
Correct
Expected:
2 4
3 6
5 3

Returned:
2 4
3 6
5 3
```

https://1o24bbs.com/t/topic/2740

k nearest point，背景是一个城市有 N 个牛排馆，牛排馆的坐标都是存在 allocations 的 list 里面(类型是 List<List<Integer>>)，然后要求返回 k 个最近的牛排馆给用户，用户位置在坐标(0, 0)。
思路：先建立每个牛排馆 location 和它到用户位置(0, 0)的距离的关系，然后用大小为 k 的 max heap 来拿到 k 个最近的位置，返回类型是 List<List<Integer>>。
这题正常做法是弄一个 inner class 来把 location 和距离的关系包装一下，我不知道为啥脑子抽了用 hashmap 来建立对应关系，最终这题 22 个 test cases 我只过了 18 个，我估摸着四个 test cases 里有重复的 location，我的 hashmap 把重复 location 滤掉了。

2. 给个无序数组要构建 BST， 然后找出 Node1 Node2 距离 我觉得这道题要好好准备一下，我准备了好久看地理有大佬 testcase 没过瑟瑟发抖中

https://www.1point3acres.com/bbs/thread-225078-1-1.html

Given a list of unique integers, construct the binary tree by given order without rebalancing, then find out the distance between two nodes.

public static int bstDistance(int[] values, int n, int node1, int node2)

for example,

values= [5,6,3,1,2,4], n is the size of values, node1 is 2, node2 is 4, then function return 3
构建完 BST 如下，2 和 4 呢，距离就是 3

```
    5
  3   6
1  4
 2
```

http://www.1point3acres.com/bbs/thread-192414-1-1.html

score gathering, 输入一个 unsorted integer array，给这个 array 打分，基本就是建 BST, 然后 BFS 输出

4, 2, 5, 5, 6, 1, 4

[num]：[出现次数]

变成

```
      4:2
  2:1      5:2
1:1          6:1
```

输出 4:2,2:1,5:2,1:1，，，6:1

如果没有节点， 输出 “”
没有孩子节点输出 “”，这样的话应该很多“”，不应该只有 1 只有有空格阿。。。
下一个 level（1:1 和 6:1）不存在就不用加空格了吧。。（2:1）没有 right，（5:2）没有 left。。就这两个空格。。

3. 棒球题目 stack 解决。这个要注意细节反正我花了好久才理解题目的意思
http://www.1point3acres.com/bbs/thread-270278-1-1.html

输入一个字符串，其中包括整数，Z，X，或者+。整数代表此轮得分，X：当前成绩是 double 前面一个分数，+：当前成绩是前两个的和，Z：移除前一个成绩，求最后的总成绩和

一颗栗子： 输入["5", "-2", "4", "Z","X", 9, "+", "+"]
output: 27
5 : sum = 5 (5)
-2 : sum = 5 - 2 = 3 (5, -2)
4 : sum = 3 + 4 = 7 (5, -2, 4)
Z : sum = 7 - 4 = 3 (5, -2)
X : sum = 3 + -2 * 2 = -1 (4 被移除了，前一个成绩是-2，-2 被 double 变成-4) (5, -4)
9 : sum = -1 + 9 = 8 (5, -4, 9)
+ : sum = 8 + 9 - 4 = 13 (前两个成绩是 9 和-4) (5, -4, 9)
+ : sum = 13 + 9 + 5 = 27 (前两个成绩是 5 和 9) 最后一个‘+’等于倒数第二个‘+’代表的分数加上 9，所以是 5+9

用一个 stack 解决。

想问楼主第一题是怎么解决的向前 out of index，比如说遇到 X，如果 stack 里只有一个数是怎么办的？还是题目有 ...
我记得题里似乎是说如果没有前面的数，就把缺失的数当作 0 来处理。时间久有点记不清了，不过题里面都有说明清楚，放心。

4. 高尔夫场砍树问题。 PQ + BFS (LC The Maze II)
http://www.1point3acres.com/bbs/thread-288738-1-1.html

golf 球场修场地。
public int flatFields (int numRows, int numColumns, List<List<Integer>> fields) {}
让小明帮公司球场修场地，给一个二维的链表 fields，场地里有坑，不能走。 场地里有树要砍掉。最后目的返回是修好一层的场地的最小步数。
Ex1:
[

[1, 3, 0, 2]
[1, 1, 3, 1]
]
上图中的 1 代表平地，可以走。 0 代表坑，不能走。 大于 1 的数字代表树木，需要砍掉。规则是从上下左右四个角开始任选一个开始走，先砍数字小的树木。 比如 2 < 3，那么就得先走 2。
上图如果从右下角开始走依次经过的坐标是： （1， 3） -> (0, 3) -> (1, 3) -> (1, 2) -> (1, 1) -> (1, 0) 所以返回的最小步数是 5， 因为通过这个路径可以修平第二层的球场[1, 1, 3, 1]， 并且走到左下角终点。
Ex2:
[
[1, 0]
[3, 2]
]
上图中的最小步数返回-1 因为，没有办法修好一层， 因为从左上角 1 开始走，不能走到 0， 也不能走 3， 因为在全局中 3 比 2 大，必须先走 2。所以就没法走了。

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=289918&highlight=amazon%2Boa

wrong answer

```
public static int cutTreeInGolf(int[][] golf) {
        // corner case
        if (golf == null || golf.length == 0 || golf[0].length == 0) return 0;
        // put all tree and corresponding point into min heap
        int row = golf.length;
        int col = golf[0].length;
        PriorityQueue<Integer> pq = new PriorityQueue<Integer>();
        HashMap<Integer, Point> map = new HashMap<Integer, Point>();
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                if (golf[i][j] > 1) { //means this point is a tree
                    pq.offer(golf[i][j]);
                    map.put(golf[i][j], new Point(i, j));
                }
            }
```

```
        }

        int res = 0;
        // get the effective element form pq one by one
        Point src = new Point(row-1, col-1);
        while (!pq.isEmpty()) {
            int treeHigh = pq.poll();
            Point tar = map.get(treeHigh);
            int currdistance = bfsFindDistance(golf, src, tar, treeHigh);
            if (currdistance != -1) {
                res += treeHigh + currdistance;
                // set the cutted tree as ground as 1
                golf[tar.x][tar.y] = 1;
                src = tar;
            } else {
                return -1;
            }
        }

        return res;
    }

    public static int bfsFindDistance(int[][] golf, Point src, Point tar, int height) {
        if(golf == null || golf.length == 0) return 0;

        int row = golf.length;
        int col = golf[0].length;
        boolean[][] visited = new boolean[row][col];
        if( !isValidGolfPlace(golf, visited, src.x, src.y, height) ) return -1;
        if( !isValidGolfPlace(golf, visited, tar.x, tar.y, height) ) return -1;
        int[] direcx = {-1,0,1,0};
        int[] direcy = {0,-1,0,1};
        PointNode root = new PointNode(new Point(src.x, src.y), 0);
        visited[src.x][src.y] = true;
        Queue<PointNode> queue = new LinkedList<PointNode>();
```

```
            queue.add(root);
            while(!queue.isEmpty()) {
                int size = queue.size();
                for(int i = 0; i < size; i++) {
                    PointNode temp = queue.poll();
                    //find a matched solution
                    if(temp.pt.x == tar.x && temp.pt.y == tar.y) return temp.dist;
                    for(int j = 0; j < 4; j++) {
                        // tranverse point is valid

                        if(isValidGolfPlace(golf, visited, temp.pt.x+direcx[j],
temp.pt.y+direcy[j], height)) {
                            int rowtemp = temp.pt.x+direcx[j];
                            int coltemp = temp.pt.y+direcy[j];
                            visited[rowtemp][coltemp] = true;
                            PointNode qadd = new PointNode(new Point(rowtemp,
coltemp), temp.dist+1);
                            queue.offer(qadd);
                        }
                    }
                }
            }
            return -1;
        }
        public static boolean isValidGolfPlace(int[][] golf, boolean[][] visited, int row, int col, int
height) {
            int m = golf.length;
            int n = golf[0].length;
            return row >= 0 && row < m && col >= 0 && col < n
                    && (golf[row][col] == 1 || golf[row][col] == height) && visited[row][col]
== false;
        }

class Point {
        int x;
        int y;
```

```
        Point(int x, int y) {
                this.x = x;
                this.y = y;
        }
}
class PointNode {
        Point pt;
        int dist;
        PointNode(Point pt, int dist) {
                //this.x = x;
                this.pt = pt;
                this.dist = dist;
        }
}
```

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=288537&extra=&page=9

```
def levelFieldTime(numRows, numColumns, field):
    # WRITE YOUR CODE HERE
    import collections
    dicts = {}
    for i in range(numRows):
        for j in range(numColumns):
            if field[i][j] > 1:
                dicts[field[i][j]] = (i,j)
    lists = sorted(dicts.iterkeys())
    def findsteps(start, end, numRows, numColums, field):
        visited = [[0 for _ in range(numRows)] for _ in range(numColums)]
        direct = [(0,1),(0,-1),(1,0),(-1,0)]
        queue = collections.deque()
        queue.append(start)
        steps = 0
        while queue:
            steps += 1
            n = len(queue)
            while n > 0:
```

```
            x,y = queue.popleft()
            for kx, ky in direct:
                i, j = x + kx, y + ky
                if (i,j) == end:
                    return steps
                if i >= 0and i < numRows and j >= 0 and j < numColumns and field[i][j] == 1
and visited[i][j] == 0:
                    visited[i][j] = 1
                    queue.append((i,j))
            n -= 1
        return -1
    ans = 0
    start = (0,0)
    for i in lists:
        end = dicts[i]
        x,y = end
        res = findsteps(start, end, numRows, numColumns, field)
        if res == -1:
            return -1
        ans += (res + field[x][y])
        field[x][y] = 1
        start = end
    return ans
field = [[1,1,0,12,1,13],
         [1,1,1,1,0,0],
         [0,1,0,0,0,0],
         [0,1,1,1,14,0],
         [0,0,0,0,1,0],
         [15,1,1,1,1,1]]
print levelFieldTime(6,6,field)
```

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=288537&extra=&page=10

```
import java.util.LinkedList;
import java.util.Map;
import java.util.Queue;
import java.util.TreeMap;

class Point {
	int x;
	int y;
	int val;

	Point(int x, int y, int val) {
		this.x = x;
		this.y = y;
		this.val = val;
	}
}

public class GolfEvent {
	public static int golf(int[][] golf, int x, int y) {
		if (golf == null)
			return 0;
		// find all trees
		int n = golf.length;
		int m = golf[0].length;

		Map<Integer, Point> map = new TreeMap<>();

		for (int i = 0; i < n; i++) {
			for (int j = 0; j < m; j++) {
				if (golf[i][j] > 1) {
					map.put(golf[i][j], new Point(i, j, golf[i][j]));
				}
			}
		}
		Point lastPoint = null, point2;
```

```
        Queue<Point> queue = new LinkedList<Point>();
        queue.offer(new Point(0, 0, golf[0][0]));
        queue.offer(new Point(0, m - 1, golf[0][m - 1]));
        queue.offer(new Point(n - 1, 0, golf[n - 1][0]));
        queue.offer(new Point(n - 1, m - 1, golf[n - 1][m - 1]));

        int steps = 0;
        for (Point p : map.values()) {
            point2 = p;
            int step = bfs(golf, queue, point2);
            System.out.println(p.x + "-" + p.y + "-" + step);
            if (step == -1)
                return -1;
            steps += step;
            queue.clear();
            lastPoint = new Point(p.x, p.y, golf[p.x][p.y]);
            queue.offer(lastPoint);// new value
        }
        queue.clear();
        queue.offer(new Point(0, 0, golf[0][0]));
        queue.offer(new Point(0, m - 1, golf[0][m - 1]));
        queue.offer(new Point(n - 1, 0, golf[n - 1][0]));
        queue.offer(new Point(n - 1, m - 1, golf[n - 1][m - 1]));

        if (lastPoint != null) {
            int rest = bfs(golf, queue, lastPoint);
            if (rest > 0) {
                steps += rest;
            }
        }
        return steps;
    }

    private static int bfs(int[][] matrix, Queue<Point> queue, Point target) {
        int n = matrix.length;
```

```
        int m = matrix[0].length;
        boolean[][] visited = new boolean[n][m];
        int[] gx = { 0, 0, 1, -1 };
        int[] gy = { 1, -1, 0, 0 };
        int steps = 0;
        while (!queue.isEmpty()) {
            int size = queue.size();
            steps++;
            for (int i = 0; i < size; i++) {
                Point p = queue.poll();
                System.out.println(":" + p.val + "x" + p.x + "y" + p.y);
                if (p.val != 1) {
                    continue;
                }
                for (int k = 0; k < 4; k++) {
                    int nx = gx[k] + p.x;
                    int ny = gy[k] + p.y;
                    if (nx < 0 || ny < 0 || nx >= n || ny >= m || visited[nx][ny] == true) {

                        continue;
                    }
                    if (nx == target.x && ny == target.y) {
                        matrix[nx][ny] = 1;// cut the tree;
                        return steps;
                    } else if (matrix[nx][ny] != 1) {
                        continue;
                    }
                    queue.offer(new Point(nx, ny, matrix[nx][ny]));
                    visited[nx][ny] = true;
                }
            }
        }
        return -1;
    }
```

5. common manager 问题 nnary-lca
http://www.1point3acres.com/bbs/thread-288537-1-1.html

https://www.geeksforgeeks.org/lca-n-ary-tree-constant-query-o1/ N-nary LCA

https://www.1point3acres.com/bbs/thread-225078-1-1.html

寻找公司员工的最低共同上司

变化 1:不是 bineary tree
变化 2:ceo 不是 manager 如果共同上司是 ceo return null

变化 3:员工可能不在这个公司 return null

6. **找所有 anagram in a String** Leetcode 438 原题

http://www.1point3acres.com/bbs/thread-288537-1-1.html

7. 水果清单 就是水果清单上的必须按顺序输出在 shoppingcart 里面
http://www.1point3acres.com/bbs/thread-288738-1-1.html

1.买水果
public static int checkWinner (List<List<String>> codeList, List<String> shoppingCart) {}
说的是小明要帮公司买水果，给了一个 codeList， 里面装的是他买的水果，给了一个 shoppingCart 里面装的是 target 水果，目标是检查 codelist 里的水果顺序是否和 shoppingCart 里的顺序匹配。注意的是只有 codelist 中的所有链表的 item 之后加起来小于等于 shoppingcart 里 item 之和才可能返回 1。 另外在 codelist 中有可能出现 item 时 "anything"， 它可以和任意的水果匹配。
Ex1:
codelist:
[
[apple, apple],
[orange, banana, orange]
]
shoppingCart: [orange, apple, apple, orange, banana, orange]
return 1, 因为 codelist 里的顺序和 shoppingcart 里除了第一个 orange 之后的水果顺序匹配

Ex2:

codelist:
[
[orange, banana, orange],
[apple, apple]
]
shoppingCart: [orange, apple, apple, orange, banana, orange]
return 0, 因为 codeList 里的顺序和 shoppingcart 没法匹配。

Ex3:
codelist:
[
[apple, apple],
[orange, banana, orange],
[pear, orange, grape]
]
shoppingCart: [orange, apple, apple, orange, banana, orange, pear, grape]
return 0, 因为 codelist 里最后一个[pear, orange, grape]中的 orange 没法和 shoppingcart 里的水果匹配。

Ex4:
codeList:
[
[apple, apple],
[orange, anything, orange]
]
shoppingCart:
[orange, apple, apple, orange, mango, orange]
return 1。

Ex5:
codelist:
[
[apple, apple],
[orange, banana, orange]
]
shoppingCart: [orange, **apple, apple**, orange, apple, **orange, banana, orange**]

return 1, 虽然[apple, apple] 和 [orange, banana, orange] 中间插入了[orange, apple]

anything 是和任何单一的水果比配，还是任意个数的水果匹配？和单一水果匹配

8，valid Parentheses LC #20 原题 stack 解 可能会加入其它不相关字符

9.给个 Movie movie BFS 找出所有关联电影中 top K rate 的电影

Set<Movie> 这道题我找了好久没有找到比较好的类题，大家将就准备
http://www.1point3acres.com/bbs/thread-225078-1-1.html

注意要抛出输入电影本身，compare rate 时候 类型是 float PQ +BFS

movie network 假设有个 Movie 类，

```
public class Movie
{
  int movieId;.
  float rating;
  List<Movie> similarMovies;
}
```

要求找和 movie 相似的电影中排名前 k 个的电影（不包括当前 movie）。就是找 movie 的所有 neighbor 中排名前 k 的电影。

https://www.1point3acres.com/bbs/thread-225078-1-1.html

题目如下：
假设有个 Movie 类，

```
public class Movie
{
  int movieId;.
  float rating;
  List<Movie> similarMovies;
还有其他的 getters
}
```

现在要求找到 k 个和 movie 最相似 的 movies。

函数的 signature 大概是这样的：

public static List<Movie> getNearest(Movie movie, int numSimilar)。

举个栗子：
m0 <-->m1, similarity 2
mo <--> m2, similarity 3
m1 <--> m3, similarity 4
m2 <--> m5, similaity 5

如果要返回和 mo 最相似的 movie, 那么应该返回 m5 (只有有一条路径从 m1 到 m5, 并且 5 是最大的）； 如果返回 3 个最相似的就返回 m2, m3， m5（顺序不重要）； 如果需要返回 10 个，但是相似的只有 9 个，那么就返回 9 个。
source movie 本身不能在返回结果里面。

可以的一个做法是 dfs + min-Heap(PriorityQueue)， 我们一直做 dfs， 每次碰到一个新的 movie，如果现在 queue 的 size 比 k 小的话，直接加到 queue 里面； 如果新 movie 的 rating 比 queue top movie 的 rating 高的话， 把顶部 movie
踢出队列，加上这个新的。

update: 应该返回 m5 (只有有一条路径从 m1 到 m5, 并且 5 是最大的） --> 应该返回 m5 (只要有一条路径从 m1 到 m5, 并且 5 是最大的)

10 菜单 就是给一个人名 list 然后菜单国家的 list，要求人名对应到国家再对应到菜品。 我的做法是建两个 hashmap
http://www.1point3acres.com/bbs/thread-280797-1-1.html

员工午饭选择推荐系统，给两个 list<string,string>, 第一个代表员工和其喜欢的菜系, 第二个代表菜系和其对应的菜，例如：
list1:
张三 中国菜
李四 美国菜
王五 日本菜
小明 *

list2:
中国菜 鱼香肉丝

中国菜 水煮鱼
美国菜 芝士汉堡

输出：
张三 鱼香肉丝
张三 水煮鱼
李四 芝士汉堡
小明 鱼香肉丝
小明 水煮鱼
小明 芝士汉堡

注意这里王五没有输出因为没有日本菜的菜单。小明 * 代表他百搭，吃什么都可以。这个题楼主用 map 和 set 做的，先遍历 list2 生成菜系和菜单表，后遍历 list1 生成推荐表。不知道这样做够不够。整个 oa 每题只有一个 default test case，其余的自己可以设计 test case 测试。

11. 最大互联点数集合 itemAssociation
http://www.1point3acres.com/bbs/thread-280797-1-1.html

http://www.1point3acres.com/bbs/thread-281940-1-1.html

亚麻卖书，每本书都有与其关联性很强的书，给一个 list<string,string> 代表书和与其关联的书，输出互相关联的最大集合。
例如：
三国演义 水浒传
水浒传 红楼梦
哈利波特 时间简史

输出：
三国演义，水浒传，红楼梦
这题楼主的解法是和在 undirect graph 找 largest connected components 的思路是一样的。

https://www.1point3acres.com/bbs/thread-281940-1-1.html

给了一些 itemAssociation, 如果一个 item 既在 association A 里面出现过，又在 association B 里面出现过，那么就把 A 和 B 合并成一个 association。求全部合并后最大的 association。
如果两个 association 一样大，返回 lexicographic order 的第一个。
input：String[][] itemAssociation
return: String[]

example:
input:
[itemA, itemB]
[itemB, itemC]
[itemD, itemE]
合并之后：
[itemA, itemB, itemC]
[itemD, itemE]
第一个有三个 item 最多，于是返回[itemA, itemB, itemC]

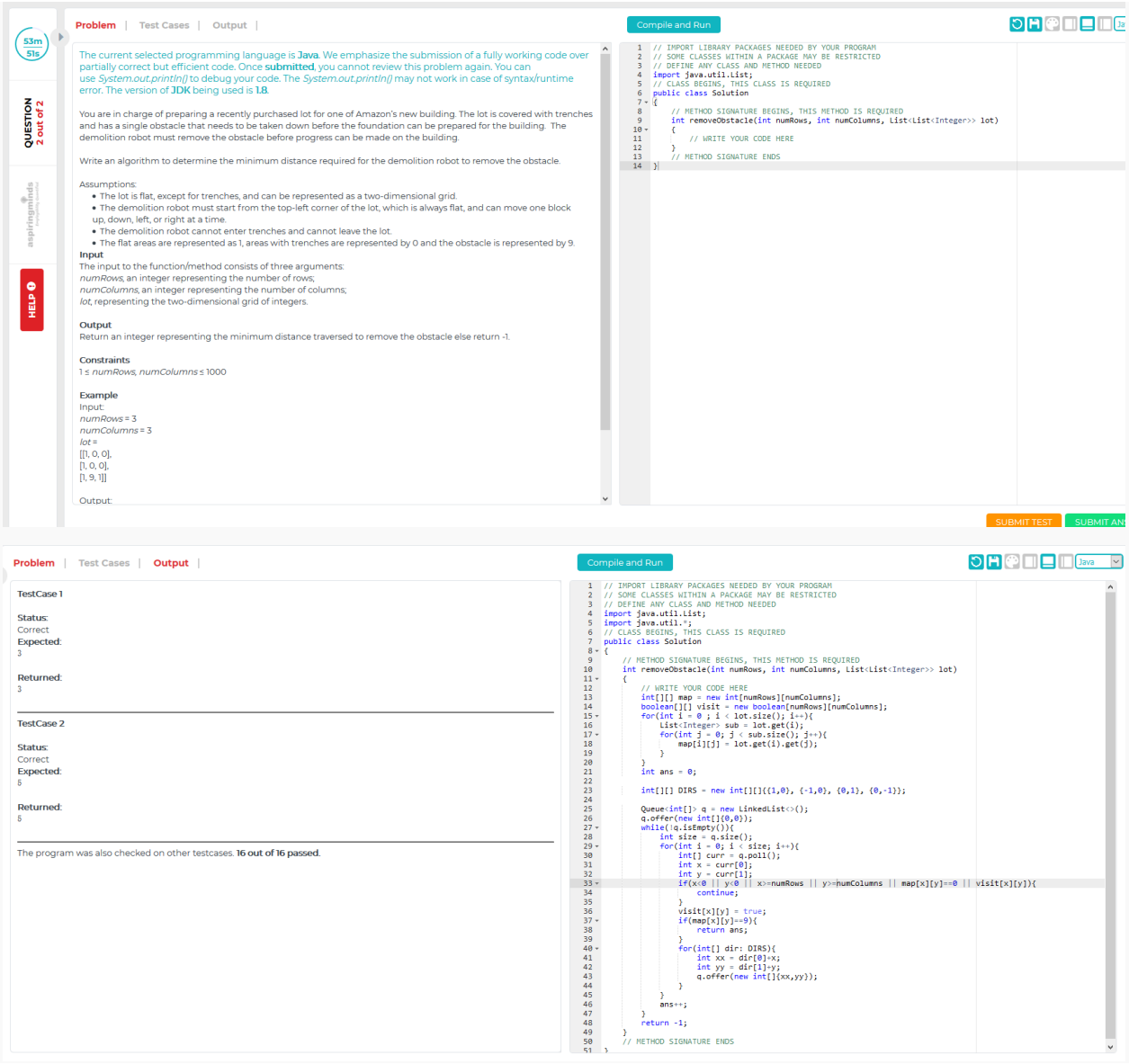
这道题我面经里没见过，现场做的。先用 UnionFind 归类，然后把最大的几个集合找出来，先对每个集合里面的 item 排序，再对集合排序，返回第一个。

12 maze 题目最近很常考
就是给一个 maze，0 不能走， 1 可以走，走到 9 问最小步数。 我的做法是用用 bfs + terminate condition

https://www.1point3acres.com/bbs/thread-225078-1-1.html

意思是说有一个 M*N 的 maze，0 代表可以通过，1 代表不可以通过，然后给你一个出口（x,y），找从（0,0）到出口的最少 steps，如果找不到 path 就返回-1
Leetcode 原题 Unique Path II

具体题目链接在这里 https://www.1point3acres.com/bbs ... 6orderby%3Ddateline



https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=479354&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

大概就是矩阵中有一个 9 是目的地，其余都是 0 或者 1，1 能走，0 不能走，怎么样最快到 9。相信大部分准备 OA 的农友都知道该怎么解，如果有实在没头绪的同学，可以看下 YouTube 这个关于 bfs shortest path 的视频，讲的十分详细，基本上源码都有了。

https://www.youtube.com/watch?v=KiCBXu4P-2Y

13 k distinct subtring

这个我开始试着用了很多的方法，都输出不对，因为输出长度也是 k， 后来用了 slidingwindow 然后 checks

14 飞机里程或者 capacity 最近， 无人机送货题

https://1o24bbs.com/t/topic/3152 amazon 全职 Amazon 社招 OA 还是之前的老 2 题

Amazon Prime Air is developing a system that divides shipping routes using flight optimization routing systems to a cluster of aircraft that can fulfill these routes. Each shipping route is identified by a unique integer identifier, requires a fixed non-zero amount of travel distance between airports, and is defined to be either a forward shipping route or a return shipping route. Identifiers are guaranteed to be unique within their own route type, but not across route types.

Each aircraft should be assigned two shipping routes at once: one forward route and one return route. Due to the complex scheduling of flight plans, all aircraft have a fixed maximum operating travel distance, and cannot be scheduled to fly a shipping route that requires more travel distance than the prescribed maximum operating travel distance. The goal of the system is to optimize the total operating travel distance of a given aircraft. A forward/return shipping route pair is considered to be "optimal" if there does not exist another pair that has a higher operating travel distance than this pair, and also has a total less than or equal to the maximum operating travel distance of the aircraft.

For example, if the aircraft has a maximum operating travel distance of 3000 miles, a forward/return shipping route pair using a total of 2900 miles would be optimal if there does not exist a pair that uses a total operating travel distance of 3000 miles, but would not be considered optimal if such a pair did exist.

Your task is to write an algorithm to optimize the sets of forward/return shipping route pairs that allow the aircraft to be optimally utilized, given a list of forward shipping routes and a list of return shipping routes.

Input
The input to the function/method consists of three arguments:
maxTravelDist, an integer representing the maximum operating travel distance of the given aircraft;
forwardRouteList, a list of pairs of integers where the first integer represents the unique identifier of a forward shipping route and the second integer represents the amount of travel distance required by this shipping route;
returnRouteList, a list of pairs of integers where the first integer represents the unique identifier of a return shipping route and the second integer represents the amount of travel distance required by this shipping route.

Output

Output
Return a list of pairs of integers representing the pairs of IDs of forward and return shipping routes that optimally utilize the given aircraft. If no route is possible, return an empty list.

Examples
Example 1:
Input:
maxTravelDist = 7000
forwardRouteList = [[1,2000],[2,4000],[3,6000]]
returnRouteList = [[1,2000]]

Output:
[[2,1]]

Explanation:
There are only three combinations, [1,1], [2,1], and [3,1], which have a total of 4000, 6000, and 8000 miles, respectively. Since 6000 is the largest use that does not exceed 7000, [2,1] is the only optimal pair.

Example 2:
Input:
maxTravelDist = 10000
forwardRouteList = [[1, 3000], [2, 5000], [3, 7000], [4, 10000]]
returnRouteList = [[1, 2000], [2, 3000], [3, 4000], [4, 5000]]

Output:
[[2, 4], [3, 2]]

Explanation:
There are two pairs of forward and return shipping routes possible that optimally utilizes the given aircraft. Shipping Route ID#2 from the forwardShippingRouteList requires 5000 miles travelled, and Shipping Route ID#4 from returnShippingRouteList also requires 5000 miles travelled. Combined, they add up to 10000 miles travelled. Similarly, Shipping Route ID#3 from forwardShippingRouteList requires 7000 miles travelled, and Shipping Route ID#2 from returnShippingRouteList requires 3000 miles travelled. These also add up to 10000 miles travelled. Therefore, the pairs of forward and return shipping routes that optimally utilize the aircraft are [2, 4] and [3, 2].

就是给了两个 list 对应编号， 让求出各拿出一个元素的和最接近一个值 我觉得这道题可以用 bfs+ priority queue 来做，但是没有动手做，地理有大佬建树做的，太大佬了， 我不是很建议用 two pointer， 因为 two pointers 要来来回回的走， 来加上重复的历程， 我用的 bf 做的。
具体题目看这个 https://1o24bbs.com/t/topic/3374

暴力解

或者我用 two pointer 优化了一下，但有些 case 过不了，麻烦老师给看下是不是逻辑问题。代码贴上了

```
import java.util.*;
```

```
public List<List<Integer>> PrimeMaxProfit(int maxTravelDist,
List<List<Integer>> forwardRouteList, List<List<Integer>> returnRouteList)
{
     List<List<Integer>> res = new ArrayList<>();
     int forLen = forwardRouteList.size(), retLen =
returnRouteList.size() ;
     if (maxTravelDist == 0 || forLen == 0 || retLen == 0) {
         return res;
     }

     Collections.sort(forwardRouteList, (a, b) -> (a.get(1) - b.get(1)));
     Collections.sort(returnRouteList, (a, b) -> (a.get(1) - b.get(1)));

     int l = 0, r = retLen - 1, diff = Integer.MAX_VALUE, sum;
     while (l < forLen && r >= 0) {
         sum = forwardRouteList.get(l).get(1) +
returnRouteList.get(r).get(1);

         if (maxTravelDist - sum >= 0 && maxTravelDist - sum <= diff) {
             if (maxTravelDist - sum < diff) {
                 diff = maxTravelDist - sum;
                 res = new ArrayList<>();
             }
             res.add(Arrays.asList(forwardRouteList.get(l).get(0),
returnRouteList.get(r).get(0)));
         }

         if (sum >= maxTravelDist) {
             r--;
         } else {
             l++;
         }
     }

     return res;
 }

public static void main(String[] args) {
    List<List<Integer>> forward = new ArrayList<>();
    List<List<Integer>> returnL = new ArrayList<>();

    forward.add(Arrays.asList(1, 5000));
    forward.add(Arrays.asList(3, 4000));
    forward.add(Arrays.asList(2, 3000));
    forward.add(Arrays.asList(4, 1000));
    forward.add(Arrays.asList(5, 4000));

    returnL.add(Arrays.asList(1, 2000));
    returnL.add(Arrays.asList(3, 5000));
    returnL.add(Arrays.asList(2, 5000));
    returnL.add(Arrays.asList(4, 6000));
```

```
    List<List<Integer>> res = new AmazonOA().PrimeMaxProfit(9000, forward,
returnL);
    System.out.println(res);
    /*Output
    [[4, 1000], [2, 3000], [3, 4000], [5, 4000], [1, 5000]]
    [[1, 2000], [3, 5000], [2, 5000], [4, 6000]]
    [[2, 4], [3, 2], [3, 3]] -> wrong! should be [[2, 4], [3, 2], [3, 3],
[5, 3], [5, 2]]
    */
}
```

你这是如果有重复的情况下出问题对吧， 不能马上 r-- 或 l++ 吧

对重复情况有问题 我也觉得是 r 或 l 那里的逻辑问题 但不知道如何 workaround 需要再加判断条件才能挪指针么

可以 probe 一下 next，如果相等的话，就把相等的全部放到 result 中。
然后把接下来是相等的全部 skip。
也就是算 sum 的时候，相等的算一个 element。但是算 result 的时候，算多个。
类似 two sum 的一段代码参考

```
while (i > 0 && nums[i - 1] == nums[i]) {
        i++;
}
```

主要是 while (nums[i - 1] == nums[i]) 的逻辑

https://1o24bbs.com/t/topic/2740

two sum closest 变形，背景是无人机送货，无人机有最大里程，然后给了两个 list，分别是出发和返回的里程数，数据类型是 List<List<Integer>>，list 里面只有 id 和里程两个值，要求找出所有出发和返回里程数之和最接近无人机最大里程的 pair。比如，最大里程 M = 11000，forwarding = [[1, 1000],[2, 7000],[3, 12000]], retrun = [[1, 10000],[2, 9000],[3, 3000],[4, 2000]], 最接近的里程和是 10000，所以结果是[[1, 2],[2, 3]].
思路：先用两个 list sort 一下，因为题目里没说给的 list 是 sort 好的，然后用 two pointers 找到最接近的里程，接着把 return list 里的 id 和里程的关系放到 hashmap 里，用 two sum 的解法就弄出来了。这题 test cases 全过了

15. 判断一个整数的二进制是否为回文数

https://www.1point3acres.com/bbs/thread-280797-1-1.html

请教楼主，二进制回文那个题，怎么算回文，是高位的 0 全都可以截断来算回文吗，还是说永远按照 32 位固定的
比如
1， 11011 （27）在高位截断下算是回文，但在 32 位固定情况下不是
2， 01000000000000000000000000000010 （536870914）在 32 位固定下算回文，但高位截断之后就不是

高位的零截断来算

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=293144&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2017(7-9 月) **码农类 General 硕士 全职@Amazon** - 网上海投 - **在线笔试** | **Other** | 在职跳槽

Tara 90min OA + BQ + Survey 无摄像头录像， 有 behavior tracking 有 web session 数量限制
1 仓库卡车运输
2 电影 top K rate 注意要抛出输入电影本身， compare rate 时候 类型是 float PQ +BFS

总结一下截止到现在**社招** OA 题库:
具体的题地里很多了 大家稍找一下就有了
1. 卡车装 M 个箱子， N 个地点 List<Integer> M<N

列出最近的 M 个位置。 PQ 解决
http://www.1point3acres.com/bbs/thread-289277-1-1.html
2. 给个无序数组要构建 BST， 然后找出 Node1 Node2 距离
http://www.1point3acres.com/bbs/thread-192414-1-1.html
3. 棒球比赛求比分问题。 stack 解决
http://www.1point3acres.com/bbs/thread-270278-1-1.html
4. 高尔夫场砍树问题。 PQ + BFS (LC The Maze II)
http://www.1point3acres.com/bbs/thread-288738-1-1.html
5. 给公司雇员结构多叉树，求两个雇员共同上司。6. 找所有 anagram
找这个同学总结的 pdf 里面有

http://www.1point3acres.com/bbs/thread-288537-1-1.html

7. 水果清单

http://www.1point3acres.com/bbs/thread-288738-1-1.html

8，valid Parentheses LC 原题 stack 解决 可能会加入其它不相关字符

9.给个 Movie movie BFS 找出所有关联电影中 top K rate 的电影 Set<Movie>

http://www.1point3acres.com/bbs/thread-225078-1-1.html

Hireon 链接 题：

10 菜单题 map + set 解决

http://www.1point3acres.com/bbs/thread-280797-1-1.html

11. 最大互联点数集合 itemAssociation

http://www.1point3acres.com/bbs/thread-280797-1-1.html

http://www.1point3acres.com/bbs/thread-281940-1-1.html

附一份准备时候用的代码草稿，大家注意要改成自己的代码，千万别用一样的，主要看个思路希望能有帮助， 攒人品 求 Onsite 另外求点米

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=484016&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 网上海投 - **在线笔试** | **Pass** | 在职跳槽

oa 其一

1. 久期三

我直接排序了没用堆也过了……

2. 机器人在 2d 平面走，从（0，0）到平面的终点，中间有壕沟不能走，算最少的步数

简单 bfs 可做

oa 其二

1. 久期三

嗯 一样的……

2. 一 list 前台程序内存占用数，一 list 后台内存占用数，还有设备的总内存：一个前台和一个后台搭配，求占用内存最多的组合（如有多个取最前的）

用 treemap 做了

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=495313&highlight=%D1%C7%C2%E9%2BOA

2019(1-3 月) **码农类 General 硕士 实习**@**Amazon** - 内推 - **在线笔试** | **Other** | 其他

第一道题是找到 number of substring containing k distinct characters。之前准备的时候，只做了对应的 k-length substring。没有考虑过长度不受限制的情况。不过，还是用 brute force，遍历所有的 substring 解决了这道题目。
第二道题是 highest five。这道题在准备的时候用了 priority_queue，其实在之前的帖子也有人说过并不能用 priority_queue，但是，我自大的没有想过这个情况。在 OA 的时候，发现不能 import 这个 library。然后自己写了一个 vector，然后每次 insert sort，最后取最高的 5 个。

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=494680&highlight=%D1%C7%C2%E9%2BOA

2019(1-3 月) **DataEng 硕士 全职**@**Amazon** - 网上海投 - **在线笔试** | **Fail** | 其他

1.新题，每次组装两个部件变为一个新部件，问组装 N 次总时间最低

2. MST minimum spanning tree
第二题没写出来，运气太背了，出了这道题
https://www.jianshu.com/p/916ae4663ab7

LC 210. Course Schedule II

LintCode 629

http://www.noteanddata.com/lintcode-629-Minimum-Spanning-Tree.html#%E9%A2%98%E7%9B%AE-lintcode-629-minimum-spanning-tree

https://www.1point3acres.com/bbs/thread-477000-1-1.html

2019(1-3 月) **码农类 General 硕士 全职**@**Amazon** - 内推 - **技术电面** | **Other** | 在职跳槽

第一题 kth closest points，用 heap 做就好了。值得说一下如果有 typo，这个编译器也不会输出任何有意义的结果，只会告诉你 return 的东西是空。debug 非常难，我就是 import heapq 写成了 heap，然后最后输出的时候又一个 type，这个题就 de 了 20 分钟。。

第二题跟这个一样 https://www.1point3acres.com/bbs/thread-476881-1-1.html
先 sort，然后倒着扫暴力解或者加个 binary search 应该都可以。这个题写了好久好久，因为我 sort array 的时候忘记指定是用 item[1]当 key，所以 sort 出来结果是错的。但是竟然过了 12 个 test

case，所以我就以为是 tle，加了 binary search，debug de 了好久，因为这个编译器基本上没有任何 debug 的功能。。。

不知道 amazon oa 可不可以先在自己的 ide 上面写好，然后再 copy 过去。。 如果可以的话，建议大家都在自己的 ide 上 debug。。

另外就是最后需要写一个 describe 你的思路和时间复杂度的 part。两道题是分开的，每个题都有两问！lz 就是做完 oa 太紧张了，傻乎乎的把两个题目都写到第一题的 page 里面，最后已经没时间写第二个 page 了。。

它提前会有一个 demo，最好先做一下。。我就是觉得一定没问题所以就 skip 了

https://www.1point3acres.com/bbs/thread-476881-1-1.html

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=476881&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 博士 全职**@**Amazon** - 内推 - **技术电面** | **Pass** | 在职跳槽

第一题 就是 K Nearest City 的题目 二维坐标的 city 找出 K 个最近的 用 priority_queue 做

第二题： Memory Consumption
其实就是大家常说的 Flight 题

有两个数组 如下形式, 每个代表一个 application, 第一个数代表 index, 代表 application 会使用的内存
[[1, 100], [2, 200], [3, 500] ...]
[[1, 200], [2, 300], [3, 900] ...]
给一个最大的 Memory 数例如 1000
从第一和第二个数组中个找出一个 application 总内存小于 1000
这个题的答案是[3, 2], 既取第一个 array 中的 3, 500 和第二个 array 中的 2, 300 总共用 800 memory < 1000

思路就是双指针 第一个头 第二个尾 第一个右移 第二个左移

第一题 就是 K Nearest City 的题目 二维坐标的 city 找出 K 个最近的 用 priority_queue 做

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=495514&highlight=%D1%C7%C2%E9%2BOA

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 猎头 - **HR 筛选 在线笔试** | **Other** | 在职跳槽

How does hashtable work? what's the time complexity? worst case?
how to sort 100,000 integers?
difference between RDBMS and nosql?

1. 有一串正整数，非排序的。先挑两个数字加起来，把这个和记下来，比如说 s1。再在剩下的列表里找一个数，加上这个和，比如说 s2. 这样直到加完所有的数字，得到 sn。最后把所有的和都加起来，s1 + s2 + ... + sn。求一种组合，使这个和最小。
比如说：
Input: 3, 1, 2
Output: 9
解释: 1+2 = 3; 3 + 3 = 6; 3+6 = 9

Input: 8, 3, 5, 2. 15
Output: 66
解释: 2 + 3 = 5; 5+5 = 10; 10 + 8 = 18; 18 + 15 = 33; 5 + 10 + 18 + 33 = 66

2 有一串字符 a-z，可能会重复。尽量把它分解成最多的 subsequence，使得一个字符不会同时出现多个 subsequence 里。返回每个 subsequence 里的字符个数
比如说，
input: ['a', 'b', 'a', 'c', 'd']
output: [3, 1, 1]

input: ['a', 'b', 'a', 'b', 'd']
output: [4, 1]

Leetcode 763

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=476062

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 猎头 - **在线笔试** | **Pass** | 在职跳槽

第一题，二维客户找餐厅，要求找最近的 n 个。客户在原点。给一堆二维坐标，再给个总共的餐厅的个数，以及 n。用长度为 n 的 max heap 搞定。需要注意的是，之前听地里说输入的坐标有可能是不合规的，所以就直接加了个 check。总觉得总共餐厅的个数给得比较多余。

第二题，飞机选路线。给两个数组，一个是去程的各个路线的里程数以及 id，一个是回程的各个路线的里程数以及 id。又给了个最长总路线 n。要求返回一个数组，是在来回路线总长度不超过 n 的前提下走最长路线的所有路线 id 组合。懒得写二分查找了，直接暴力双循环也给过了

饭店是旧漆散

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=466482&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=473852&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 博士 全职@Amazon** - 内推 - **在线笔试** | **Other** | 在职跳槽

题目： 1， K nearest locations。应该是领扣陆毅儿 2， closest target value， 领口意思起吧很想。

我是社招，走社招的同学还是要注意点区别的：
1， 社招 OA 只有 90 分钟，两道题，虽然远远够用，但是也要注意时间。
2， 要问时间复杂度和解题思路。
3， Java 的语言要稍微熟悉下，因为亚麻用 java，所以其他语言有可能会有些 case 过不了。java 把 list，prioryQueue，还有 hashMap 的 library 熟悉下，基本上就无忧了。
4， 不能直接套用领口或者利口的逻辑和方法，因为题目和传参不一样。
5， 感谢这个帖子给了我很大帮助，也建议各位同学准备前看一看

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=473879&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Other** | 在职跳槽

一道 two pointer less or equal 一道 k nearest 都是地里的题

> 可以说一下第一题 two pointer less or equal 具体是怎样的吗?

就是两个数组， 每个数组取一个, 求两数之和不大于 target 的最大组合

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=475294&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Other** | 在职跳槽

1. PrimeNow K 个最近的送货点
2. 前后台程序找 pairs，使得相加的 mem consumption 最接近给定容量值。也就是先找最接近的值，然后列出所有符合这个值的 pairs。

work style 同校招 OA1

请问第二道题可否有代码可以参考?
我当初用 two pointer 有三个 test case 没过，想不出错在哪

不是很明白为什么会用到 two pointers，两层 for-loop 就行了……

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=473276&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2019(1-3 月) **码农类 General 本科 全职@Amazon** - 猎头 - **技术电面** | **Other** | 在职跳槽

都是高频面经题，不难，半个小时写完两题
1. Top k closest restaruant, 用 maxheap 做就行，注意，test case 从原来的 15 个增加到 22 个了，我只过了 21 个，我 debug 了半个小时，也没找到另外一个 case 怎么 fail 的，放弃了...2. 飞机来回路程问题，类似 two sum closest, 按照之前面经说的直接暴力 n^2 算法做就好，test case 很容易全过

最后传送一个其他人的面经总结:
https://www.1point3acres.com/bbs ... read&tid=461770

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=461770

2018(10-12 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Pass** | 在职跳槽

1. K 临近点变种 。常见的用 heap 做（领扣 612 基本一样）
2. 2D 矩阵迷宫：0 能走，1 走不通，9 终点。找从（0，0）开始到 9 终点的最短路径。BFS 即可（领扣 1092 会用到，做通这题就好)

另外还有总结了近期的题库，以下是七道我找到领扣里面最接近的题号：

612, 1092, 386, 384, 1478, 1561, 1369

注意以上题号是领扣的不是李扣的。好好把上面几题做一遍可以保证基本的自信心。。

同时翻译一下楼主的 lintcode 题号 到 leetcode 题号：

lintcode 612 -> leetcode 973(similar but easier)
lintcode 1092 -> leetcode 675
lintcode 386 -> leetcode 340
lintcode 1478 -> ?
lintcode 1561 -> leetcode 783(similar but easier)
lintcode 1369 -> leetcode 819

2018(10-12 月) **码农类 General 硕士 全职@Amazon** - 网上海投 - **在线笔试** | **Other** | fresh grad 应届毕业生

流程依旧是

- a coding challenge with two scenarios (**up to 90 min**)
- a "describe your approach" section to discuss your coding solutions (**up to 15 min**)
- a work style survey (**up to 15 min**)
- a feedback survey (**5 min**)

描述思路的时候脑残以为没有字数限制，结果写超了 68 个字母 到时间了，也不知道发生了什么，写超了还不能提交，不知道有没有默认帮我保存一个版本，没有的话就哭哭惹。小伙伴们要留神，虽然感觉应该没有我这么蠢的。。。。

1.一堆蔬菜店（牛排店）的坐标，给了个原点（0，0），求 k 个距离最近的店， 等同于林扣六一二（缩水版，因为这道题 oa 题把原点坐标定死了反而操作少了些） 2.matrix 迷宫，0 是墙，1 是路，9 是目标点，从左上出发，上下左右都可以，问到目标的最短路径是多少，没有就 return-1 （搜这个就好了 shortest path in a binary maze）貌似还有一道可能的题 没碰到 devicecapacity input 是 front=[[1,3000],[2,5000],[3,7000],[4,10000]] back=[[1,2000],[2,3000],[3,4000],[4,50000]] ,给一个最大的

capacity 限制例如 10000， 然后返回所有符合限制组合中 capacity 最大的所有组合，output[[2,4],[3,2]] ,搜集到的解法貌似暴力走也行 双指针也行 感觉可能和双指针里的 closet sum 有点像

具体细节就搜 https://www.geeksforgeeks.org/shortest-path-in-a-binary-maze/

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=461528&extra=&page=1

2018(10-12 月) **码农类 General 硕士 全职@Amazon** - Other - **在线笔试** | **Other** | 在职跳槽

1. Amazon 想给 k 个地点送货，出发点是(0,0)，就是 K nearest point 的变形。给 List<List<Integer>> location，number of deliveries K. 我用的 priority queue
2. 一个 M*N 的 matrix，1 为可以走，0 为不能走，9 为 destination。求最短步数。我用的 bfs

总之都是地里的原题，还有 15 分钟 describe your approach 和分析 complexity。 一个 work style survey。

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=443853&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2018(7-9 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Pass** | fresh grad 应届毕业生

上周五拿到 OA，周末准备了下，昨天把 OA 做了。做的两道题都是题库里的题，面的 alexa 组。1. k nearest point，背景是一个城市有 N 个牛排馆，牛排馆的坐标都是存在 allocations 的 list 里面(类型是 List<List<Integer>>)，然后要求返回 k 个最近的牛排馆给用户，用户位置在坐标(0, 0)。
思路：先建立每个牛排馆 location 和它到用户位置(0, 0)的距离的关系，然后用大小为 k 的 max heap 来拿到 k 个最近的位置，返回类型是 List<List<Integer>>。
这题正常做法是弄一个 inner class 来把 location 和距离的关系包装一下，我不知道为啥脑子抽了用 hashmap 来建立对应关系，最终这题 22 个 test cases 我只过了 18 个，我估摸着四个 test cases 里有重复的 location，我的 hashmap 把重复 location 滤掉了。

2. two sum closest 变形，背景是无人机送货，无人机有最大里程，然后给了两个 list，分别是出发和返回的里程数，数据类型是 List<List<Integer>>，list 里面只有 id 和里程两个值，要求找出所有出发和返回里程数之和最接近无人机最大里程的 pair。比如，最大里程 M = 11000，forwarding = [[1, 1000],[2, 7000],[3, 12000]], retrun = [[1, 10000],[2, 9000],[3, 3000],[4, 2000]], 最接近的里程和是 10000，所以结果是[[1, 2],[2, 3]].
思路：先用两个 list sort 一下，因为题目里没说给的 list 是 sort 好的，然后用 two pointers 找到最

接近的里程，接着把 return list 里的 id 和里程的关系放到 hashmap 里，用 two sum 的解法就弄出来了。这题 test cases 全过了。

昨天做完 OA 后心情很差，第一题有四个 test cases 没过，感觉是吹了，结果下午 hr 给我来了个电话，通知 onsite，看来 test case 没全过也是有机会的。

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=350941&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0

2018(1-3 月) **码农类 General 硕士 全职@Amazon** - 猎头 - **在线笔试** | **Other** | fresh grad 应届毕业生

【亚麻社招】新鲜出炉 Amazon 两道 OA 新题

第一题是词频题，给一个 String，然后一个 exclusiveWordList (List<String>)，找出 String 里去掉 exclusive word 之后的最高频词汇的 list
第二题是 log file.
详情请看：
http://www.1point3acres.com/bbs/thread-348589-1-1.html

查找最大词频的题，用个 Map 记录词频，然后排序输出肯定是过不去的

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=330227

2018(1-3 月) **码农类 General 硕士 全职@Amazon** - 内推 - **在线笔试** | **Other** | 在职跳槽

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313308&highlight=%D1%C7%C2%E9%2Boa%2B%C9%E7%D5%D0%2B%C3%E6%BE%AD

2017(10-12 月) **码农类 General 硕士 全职@Amazon** - 猎头 - **Onsite** | **Fail** | 在职跳槽

先是 90 分钟 OA，是版上常见的水果题，和构建 BST 然后找两个 node 之间的最短距离