亚马逊OA

Round Robin: <https://www.youtube.com/watch?v=aWlQYllBZDs>

一道是Valid parentheses   
不知道为什么 有三个test case怎么都过不去。。。。  
  
还有一道是bst 找distance的题   
signiture 大概是  
public static int bstDistance(int[] values, int n, int node1, int node2).1point3acres缃�  
  
做完coding 是15min的一个什么测试

1. 水果题  
开始在找机经的时候发现有些人在问 水果codelist中间 能不能插入其他水果，这里是不允许的；另外，anything也必须match一个水果，不能多个或者没有，所以不是wildcard, 其实很好做，loop一遍codelist就行了  
2. movie network  
输出N个最高评分的关联电影  
BFS + min heap of N 搞定

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。.1point3acres缃�  
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:. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
[  
[apple, apple],. Waral 鍗氬鏈夋洿澶氭枃绔�,  
[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: . 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
codelist:  
[  
[orange, banana, orange]，  
[apple, apple]. from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
]. from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
shoppingCart: [orange, apple, apple, orange, banana, orange]-google 1point3acres  
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。. Waral 鍗氬鏈夋洿澶氭枃绔�,  
  
Ex4:  
codeList:  
[  
[apple, apple],  
[orange, anything, orange]  
]. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
shoppingCart:. 1point 3acres 璁哄潧  
[orange, apple, apple, orange, mango, orange]  
return 1。

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。. more info on [1point3acres.com](http://1point3acres.com/)  
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].1point3acres缃�  
[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。

第一题：shopping list和basket.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
给一个shopping list 比如说 [[apple, banana], [orange, apple], [anything, orange]].鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
然后checkbasket的买东西是不是match shopping list.  里层数组的东西和顺序必须match，数组与数组之间可以插入任何东西， anything表示任何东西都可以买  
比如basket [apple, banana, apple, apple, orange, apple, banana, orange, apple, orange]就是match的，return 1  
  
第二题：跟这里的第二题一样  <http://www.1point3acres.com/bbs/thread-288537-1-1.html> 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
先用pq按顺序保存要砍的树，然后bst找到每棵要砍的树，注意BST找路径的时候condition是check if height ==1，但是到了目标树它的高度其实是 > 1的，所以bst在达到目标树之前就会停止，导致找不到路径。 所以要先把目标取出来，然后把树砍了（set height = 1），然后再用BST找。 LZ就是把这两句写反了，没时间改了。。。。。时间都花在修我家的破网和解决comparator generics的compare method必须要public但是solution的method是protected上面了。。。哭。。。  
第二题的树的高度好像没有相等的，如果要考虑树的高度相等的情况，那就很麻烦了。

[Amazon](http://https//www.amazon.com/b?_encoding=UTF8&tag=1point3acres-20&linkCode=ur2&linkId=89c11e2c5b86155c5422f19cca1e9880&camp=1789&creative=9325&node=5) warehouse。。。其实就是给你x,y 然后算x,y 到原点的距离，输出最小的几个，java应该priorityqueue就够了，我用的python，也还可以。  
第二题，golf event要砍树。。。每次只能砍所有树里面最矮的那颗。其实就是maze题的变形。2D-array. 0不能走，1可以走，>1 就是树，要求的输出就是从原点开始，走到每颗当前树里面最矮的那颗所需的步数+需要砍得树的高度的总和。方法我就是先找好所有的树，排好序，然后从一个点到另一个点做BFS。 找出最小步数。. Waral 鍗氬鏈夋洿澶氭枃绔�,  
举个例子 [[1,1,0,2],[3,1,1,1]], 从（0，0）走到 （0，3）--》2 这棵树，就是5步+2（树高），然后从（0，3）走到 （1，0）->3 这棵树 4步+3（树高）所以5+2+4+3返回14

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:. Waral 鍗氬鏈夋洿澶氭枃绔�,  
            steps += 1  
            n = len(queue)  
            while n > 0:  
                x,y = queue.popleft(). 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
                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. Waral 鍗氬鏈夋洿澶氭枃绔�,  
        return -1  
    ans = 0.1point3acres缃�  
    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. Waral 鍗氬鏈夋洿澶氭枃绔�,  
        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],. [1point3acres.com/bbs](http://1point3acres.com/bbs)  
         [15,1,1,1,1,1]]. From 1point 3acres bbs  
print levelFieldTime(6,6,field)

1. 水果清单

刚开始有三四个testcase没过  
\* 不要忘记写anything的情况.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
\* 如果codeList 和 shoppingCartList都为空时，应该返回1。  
修改后就AC了。

codelist:  
[  
[apple, apple],   
[orange, banana, orange]  
]  
shoppingCart: [orange, apple, apple, lychee, orange, banana, orange]  
  
分别把 codelist里的东西按顺序匹配在shopping cart 中  
在shopping cart 中， 一个code list里的东西不能分开。红的里是codelist[0], 橙色里是codelist[1]。它们中间隔了一个lychee, 这时允许的。 同时，在code list里的组合在shopping cart 中必须依次出现， 顺序不能打乱。  
  
  
思路：   
1. 在shopping cart中找到匹配的codelist[0], 这个类似<https://leetcode.com/problems/implement-strstr/>， 假设找到的字符串在shopping cart 中的结束坐标为 index,. 鍥磋鎴戜滑@1point 3 acres  
2. 从 index + 1开始找codeList[1]  
3. 重复上述过程，直到把codeList里面的元素都找到返回1,  
4. 如果已经搜索到 shopping cart结尾，仍未找到codeList里所有元素， 就返回0

2. 给一个无序数组，构建bst，找出给的距离

首先查看给的两个node是否在数组里都能找到，如果不能则直接返回-1  
接着写几个函数：. From 1point 3acres bbs  
\*通过每次从root查找建BST,  
\*找LCA，返回该lca node,  
\*计算LCA node 与node1距离，计算LCA node 与 node2距离。     
\*返回距离和

第一个题是求到给定点的最近的k个点的问题吗

第一题是k closest point吧 lincode原题  
1. 请问第二题可以走有数的地方么？ 2. 树被砍了之后是默认这个地方为1 还是 0？ 3. 可能出现一个多颗高度一样的树么？  
1.不可以 2.砍完是1 3.不会

golf球场砍树  
  
<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=291981&pid=3149745&page=1&extra=page%3D1%26filter%3Dsortid%26sortid%3D311#pid3149745>

链接: <https://instant.1point3acres.com/thread/278257>  
来源: 一亩三分地

\* 你有没有在时限很短的情况下完成一个项目，你是怎么调节你的时间的，你是否因此必须牺牲什么？ \* 有时候碰到deadline快到，但是项目遇到问题有可能做不完，怎么处理. \* 你最proud of的一个项目是什么 \* 你最成功deliver project的经验

你有没有没赶上DeadLine的项目 Could done better Describe a situation that you have done a lot on the project, but you find your method for the problem is not good, how would you deal with that ? how you deal with a coworker who dont like you, who you dont like Build relationship with co-worker Why amazon?

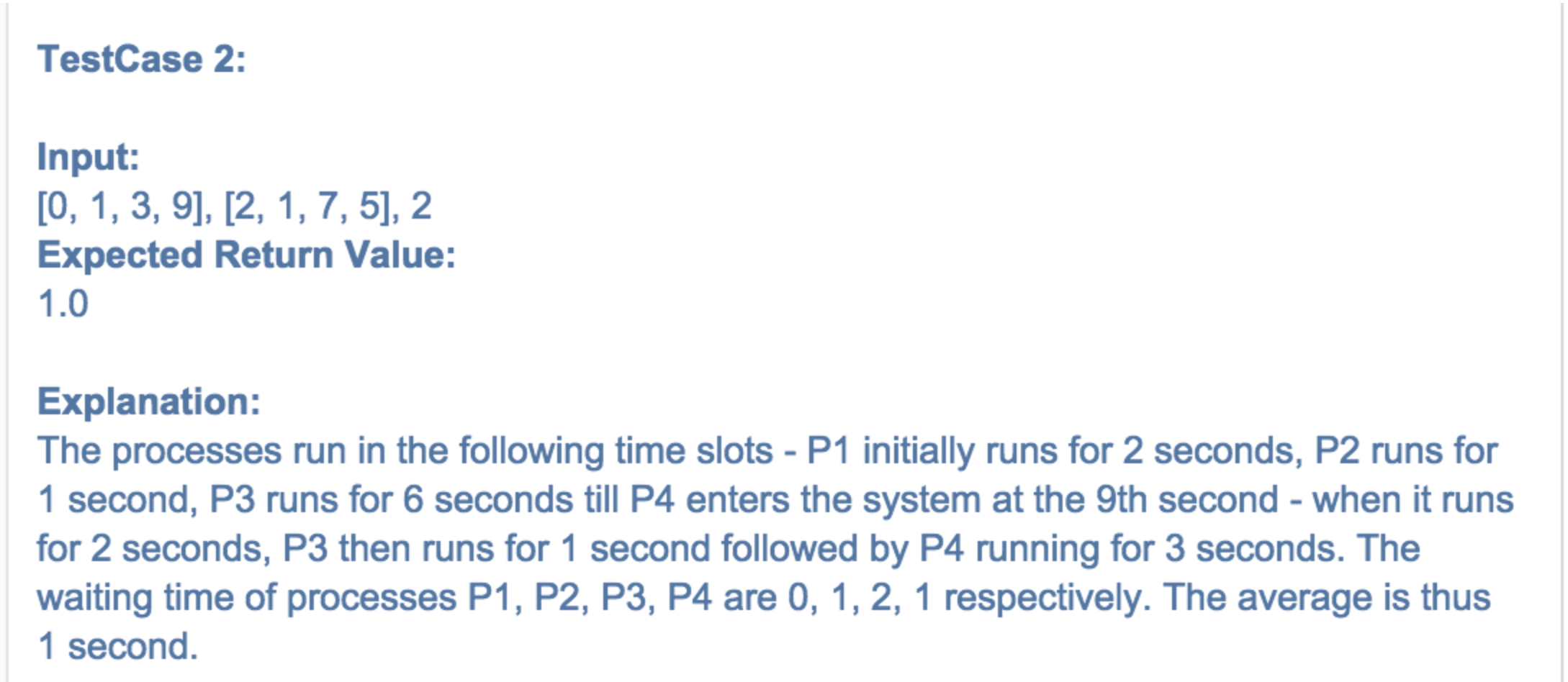
链接: <https://instant.1point3acres.com/thread/277830>  
来源: 一亩三分地

重点来了，之前的coding好像只有4选2，我碰到了2个不一样的！！ 第一题是winsum 变形， ［1，2，3，4，5］ k = 2, 输入每个window里小的那一个， 也就是说输出［1，2，3，4］ 

就是给一个list ［1，2，3，4，5］ 和一个window size， 输出每一个window里较小的那一个， 不如第一个window ［1，2］， 小的那个是1， 所以输出的list里第一个数是1，以此类推  
  
链接: <https://instant.1point3acres.com/thread/287052>  
来源: 一亩三分地

第二题ITEM ASSOCIATION。 input , 表示物品A和物品B相互关联。 , 表示物品B和物品C相互关联。 如果物品相互关联，就组成一个组。最后要求找出物品最多的那个组。 不难但是最后有个测试用例一直过不了。三天后还是收到EMAIL，这周五ONSITE.

我就用list> 把组都存起来。来一个新pair: itemA, itemB,就扫描一下看看是不是分别已经和以后的组关联。如果AB关联不同组，需要把组合并。否则加到已有组或者新建一个组ssa



|  |
| --- |
| 我是这么理解的，有不对的地方请指出. 1point 3acres 璁哄潧 我拿贴子里面给出的例子来说明下吧。. [1point3acres.com/bbs](http://1point3acres.com/bbs) 【0，1，4】 【5，2，3】 q ＝ 3  第0秒，任务1进队列。我们peek目前队列中有的任务是任务1，我们发现任务1开始时间第0秒，目前也是第0秒，所以任务1等待时间是0。然后任务1执行3秒，它自身还有2秒钟。 这时候我们查看3秒的时候哪些任务达到了，发现任务2在第1秒到达。于是任务2进队列。 这时候我们查看任务1有没有执行完，发现没有执行完，于是我们poll任务1，再把任务1 add到队列末尾。 这时候队列的顺序是任务2，任务1. . [1point3acres.com/bbs](http://1point3acres.com/bbs) 现在我们再次peek队列，于是找到任务2.我们发现任务2在第1秒到达了，目前我们在第3秒。所以等待时间是3-1=2. 我们重复刚刚的步骤，发现任务2执行时间只要2秒，于是我们到第5秒。这时候我们查找第5秒哪些任务到达了。我们发现任务3也到达了。于是任务3进队列。. more info on [1point3acres.com](http://1point3acres.com/) 所以目前队列顺序是任务2，任务1，任务3. 我们又发现任务2已经执行完了，于是  以下内容需要积分高于 155 才可浏览  我们把任务2 poll出队列，不再把它放进队列里了。 所以现在队列里面剩余的任务是任务1，任务3.  于是我们再peek队列。请注意，这时候的q被重新设置过了，不是3－2=1秒，而是又是3秒。. [1point3acres.com/bbs](http://1point3acres.com/bbs) 我们再次peek队列，找到任务1，目前是在第5秒，我们刚刚执行过任务1，他暂停在第3秒，所以任务1又等了2秒。目前秒数是2+2=4秒。 目前任务1还有2秒。我们执行完任务1以后，到达第5+2 ＝ 7把他扔出队列。目前队列里只有任务3了。 -google 1point3acres 然后我们再peek，现在只有任务3了，目前我们在第7秒。任务3进来的时候在第4秒。所以任务3等了7-4 ＝ 3秒。  所以等待时间又加3秒。 所以最终等待时间是2+2+3 ＝ 7秒。平均等待时间是7/3 ＝ 2.3333秒。 |

public class process {  
        int arriveTime;  
        int excuteTime;  
        process(int arr, int exc) {  
                arriveTime = arr;  
                excuteTime = exc; 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
        }  
}  
  
// Assume arrive is sorted.  
public double roundRobin(int[] arrive, int[] excute, int q) {  
        LinkedList<process> queue = new LinkedList<>();  
        int curTime = 0; 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
        int waitTime = 0;  
        int nextProIdx = 0;.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
        while (!queue.isEmpty() || nextProIdx < arrive.length) {. 1point 3acres 璁哄潧  
                if (!queue.isEmpty()) {  
                        process cur = queue.poll();  
                        waitTime += curTime - cur.arriveTime;  
                        curTime += Math.min(cur.excuteTime, q);  
                        for (int i = nextProIdx; i < arrive.length; i ++) {  
                                if (arrive[i] <= curTime) {  
                                        queue.offer(new process(arrive[i], excute[i]));  
                                        nextProIdx = i + 1;  
                                } else {  
                                        break;  
                                }  
                        }  
                        if (cur.excuteTime > q) {  
                                queue.offer(new process(curTime, cur.excuteTime - q);. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
                        } 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
                } else {  
                        queue.offer(new process(arrive[nextProIdx], excute[nextProIdx]));  
                        curTime = arrive[nextProIdx ++];. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
                }  
        }  
        -google 1point3acres  
        return (double)waitTime / arrive.length;  
)