K-1 distinct subarray with length K  
多差树 找最大平均的TreeNode 不包括叶子om 1point 3acres bbs  
  
变形题：  
Number of substrings containing K distinct characters，返回满足条件的子串个数  
    -corner case：K=0时，返回0  
    -注意相同的子串只算一次，所以用unordered\_set来记录符合条件的子串，可以达到除重的目的  
  
**若要求子串长度为K：**  
    -可以考虑用双指针来模拟大小为k的窗口，用一个unordered\_map来存窗口内的，移动时把左指针所指的字符在map中—，若等于0则从map中删去该项；而右指针所指的新字符则在map中++（或新建），若map.size等于k则说明该子串符合条件，否则不符合条件。  
    -时间复杂度为O(n+mk)其中n为大字符串的长度，m为符合条件的子串数目（mk是取substr的开销）  
  
**若子串长度没有限制：**  
    -举例来说，“aba”的k=2的子串有“ab”、“aba”，亦即要求有k种字母，而单种字母是可以出现多次的  
    -用两重循环的方法，对每个左指针i，右指针j从i开始一直移动到n-1，此过程中维护一个hashmap。若hashmap的size为k，则将该子串加入到结果中  
    -时间复杂度是O(n^2)。如果相同子串只算一次的话，需要用unordered\_set来存结果。需要注意的是每一层i循环中移动指针j的时候也需要维护一个[i,j]的子串，这样当符合条件时就不需要调用substr产生额外的开销了  
  
  
**find substrings of size k with k-1 distinct characters，返回满足条件的子串个数**  
    -与上题算法的区别：判断条件从map.size==k改为map.size==k-1

**K closest point to origin**

**Time complexity: O(nlogk), where n is the number of points.**

**Space complexity: O(k). A PriorityQueue of size k is used.**

**public Point[] kClosest(Point[] points, Point origin, int k) {**

**Queue<Point> pq = new PriorityQueue<>(k + 1, new PointByDistToOriginComparator(origin)); // max pq**

**for (Point point : points) {**

**pq.add(point);**

**if (pq.size() > k) pq.poll();**

**}**

**Point[] result = new Point[pq.size()];**

**for (int i = pq.size() - 1; !pq.isEmpty(); i--) {**

**result[i] = pq.poll();**

**}**

**return result;**

**}**

**private static class PointByDistToOriginComparator implements Comparator<Point> {**

**private final Point origin;**

**public PointByDistToOriginComparator(Point origin) {**

**this.origin = origin;**

**}**

**public int compare(Point p1, Point p2) {**

**int dist1 = distanceToOrigin(p1);**

**int dist2 = distanceToOrigin(p2);**

**int cmp = Integer.compare(dist2, dist1);**

**if (cmp == 0) {**

**cmp = Integer.compare(p2.x, p1.x);**

**}**

**if (cmp == 0) {**

**cmp = Integer.compare(p2.y, p1.y);**

**}**

**return cmp;**

**}**

**private int distanceToOrigin(Point p) {**

**int dx = p.x - origin.x;**

**int dy = p.y - origin.y;**

**return dx \* dx + dy \* dy;**

**}**

**}**

**Shortest distance to 9 in matrix**

**public int shortestDistance(int[][] maze, int[] start, int[] dest) {**

**int[][] distance = new int[maze.length][maze[0].length];**

**for (int[] row: distance)**

**Arrays.fill(row, Integer.MAX\_VALUE);**

**distance[start[0]][start[1]] = 0;**

**int[][] dirs={{0, 1} ,{0, -1}, {-1, 0}, {1, 0}};**

**Queue < int[] > queue = new LinkedList < > ();**

**queue.add(start);**

**while (!queue.isEmpty()) {**

**int[] s = queue.remove();**

**for (int[] dir: dirs) {**

**int x = s[0] + dir[0];**

**int y = s[1] + dir[1];**

**int count = 0;**

**while (x >= 0 && y >= 0 && x < maze.length && y < maze[0].length && maze[x][y] == 0) {**

**x += dir[0];**

**y += dir[1];**

**count++;**

**}**

**if (distance[s[0]][s[1]] + count < distance[x - dir[0]][y - dir[1]]) {**

**distance[x - dir[0]][y - dir[1]] = distance[s[0]][s[1]] + count;**

**queue.add(new int[] {x - dir[0], y - dir[1]});**

**}**

**}**

**}**

**return distance[dest[0]][dest[1]] == Integer.MAX\_VALUE ? -1 : distance[dest[0]][dest[1]];**

**}**

* 1. **K distinct elements**

变种1：K length K distinct

解法： <https://www.lintcode.com/problem/k-substring-with-k-different-characters/note>

if (stringIn == null || stringIn.length() == 0 || K > stringIn.length()) {

return 0;

}

Set<Character> charSet = new HashSet<>();

Set<String> resSet = new HashSet<>();

for (int i = 0; i < stringIn.length(); i++) {

for (int j = i; j < stringIn.length(); j++) {

if (!charSet.add(stringIn.charAt(j))) {

charSet.clear();

break;

}

if (j == i + K - 1) {

resSet.add(stringIn.substring(i, j + 1));

charSet.clear();

break;

}

}

}

return resSet.size();

变种2：K length K-1 distinct

解法： <https://www.lintcode.com/problem/k-substring-with-k-different-characters/description>

<https://www.1point3acres.com/bbs/thread-476958-1-1.html>

Problem I Test Cases 'l Output •l 
Michelle has created a word game for her students. The word game begins with 
Michelle writing a string and a number, K, on the board. The students must find a 
substring of size K such that there is exactly one character that is re—eated once; in 
other words, there should be K —h distinct characters in the 
Write an algorithm to help the students find the correct answer. If no such 
substring can be found, return an empty list; if multiple such substrings exist! 
return all of them, without repetitions. The order in which the substrings are 
returned does not matter. 
Input 
The input to the function/method consists of two arguments - 
inputString, representing the string written by the teacher; 
num, an integer representing the number, K, written by the teacher on the board. 
Output 
Return a list of all substrings of inputString with K characters, that have K-1 distinct 
character i.e. exactly one character is repeated, or an empty list if no such substring 
exists in inputString. The order in which the substrings are returned does not 
matter. 
Constraints 
The input integer can only be greater than or equal to O and less than or equal to 
26 (0 num 26) 
The input string consists of only lowercase alphabetic characters. 
Examples 
Input: 
inputString awaglk 
num 4 
Output: 

变种3：K distinct

<https://www.geeksforgeeks.org/count-number-of-substrings-with-exactly-k-distinct-characters/>

N Nettiix 
M Amaten X Aspirng Minds 
amazon 
Automata 
x Queue Interface In Java-C X How do' declare andhtia 
Compile and Aun 
i 
The current *lected programming language is Java We emphasize the 
submrwon of a fully working code over part.ally correct but efftc.ent code. 
submitted. you cannot rev•e•.•. trus problem again. Vou Can 
use System our mntln() to debug your code The Systemout.print/nt) may not 
In case of syntaxfrunt'rne error The version of JDK being used vs 1.8 
Wily Wonka devised a fun game for the CRY to help identiW deserving visitors to his 
Factory tn City and the gar'* using 
advertisements in newspapers distributed all around the city The public has to find the 
With exactly •K• distinct characters 
Write an algonthrn to F*/p a deserving Visitor find the right solution if ttw goen string 
dc— not have K distinct characters return O. 
input to tt--,e furrti0r*neth0d consists Of two argurnents 
inputStnng reptewntng the String provided in the adverttsernent• 
nu.•n integer the K. prm&d in the 
an integer repregnting the Of With exactly distinct 
cha 
The integer can only greater than 0' equal 
• 26 (O 
nurn 261 
The in mn String charEters 
The dent" i ed substrings need to diqinct 
.•npurStnng 
Exolanation: The followina are the substnnas with exaeuv a distinct cha' acters 
4 
5 
6 
7 
8. 
9 
11 
12 
13. 
14 
17 
18 
19 
21 
22 
23 
24 
25 
26 
28 
29 • 
30 
31 
32 
33. 
34 
35 
36 
// LIBRARY PACKAGES NEEDED BY PROGR.* 
// CLASSES WITHIN A PACKAGE BE RESTRICTED 
// DEFINE ANY CLASS NEEDED 
// CLASS BEGINS. THIS CLASS IS REWIRED 
java. util. • 
public class Solution 
// SIGNATURE BEGINS, THIS IS REWIRED 
int ca.mtKDistinctS'bstrings(Strirv inputStriry, int rx.) 
// MITE HERE 
if(inputString — null I inputString. length() 
return e; 
if(run — 
return O; 
1 
int r•esutt — O; 
int 
freq ne intC26J•; 
forCint e; 
Arrays. f it I(freq. e) ; 
for(int j 
char cur charAt(j) 
result.. ; 
MacBook Pro 

* 1. **Closet two sum**

<https://wdxtub.com/interview/14520850399861.html>

<https://www.1point3acres.com/bbs/thread-478916-1-1.html>

2 coding challenges (70 mins): 
a. closest two sum < target, return pair 
my thought process: 
1 
2 
3 
4 
5 
6 
7 
8 
9 
10 
11 
12 
13 
14 
15 
16 
17 
18 
19 
20 
21 
22 
23 
24 
25 
26 
// CLOSeS1 two sum k 
class Solution { 
public int [l numbers, 
Arrays. sort (numbers) ; 
int I 0; 
Int h 
numbers. length 
int d 
Integer. MAX _ VALUE; 
int [J 
new int [21; 
res 
(1 
while 
int target) { 
numbers [l] 
numbers [h] ; 
int cur — 
int diff 
ta rget 
cur; 
(diff 
diff 
if 
d 
diff; 
res [01 
numbers [l] ; 
res [Il 
numbers [h] ; 
(cur 
< target) { 
if 
else 
return 
res;

<https://www.1point3acres.com/bbs/thread-478168-1-1.html>

1. two sum closest(<= target) 
Input: 
the input to the function/method consists Of three arguments, 
numContainers, an integer representing the number of containers; 
maxCapacity, a real number representing the maximum capacity of the shelf; 
containerWt, a list of rea/ numbers represneting the weight of the containers 
Output: 
Return a pair of real numbers representing the weights of the two heaviest containers chosen to be carried on the shelf. 
Example: 
Input: 
num Containers 6 
maxCapacity= 433 
103.7313, 11.2423.79, 18.350] 
Output: 
(23.79 18.350/ 

# 

* 1. **Max average subtree**

本地vscode： <https://www.lintcode.com/problem/subtree-with-maximum-average/description>

<https://www.1point3acres.com/bbs/thread-478916-1-1.html>

b. max average sum subtree (general tree, leaves not counted), return tree node 
import java.util..; 
public class Solution 
class Result { 
CategoryNode node; 
int sum; 
int size; 
public Result(CategoryNode node, 
this. node 
node; 
this. sum - sum 
this.size - size 
private Result res null; 
int sum, 
int size){ 
// METHOD SIGNATURE BEGINS, THIS METHOD IS REQUIRED 
pub tic CategoryNode getMostPopularNode(CategoryNode rootCategory) 
( rootCategory 
if 
return null; 
Result rootRes : helper( rootCategory) ; 
res. node; 
return 
// METHOD SIGNATURE ENDS 
public Result helper(CategoryNode root){ 
if 
( root 
return new Resutt(nult, e, e); 
neu 
results - 
for 
int 
int 
for 
(CategoryN0de n : root. 
results. add (helper 
childSum " 
childSize 
(Result r: 
childSum 
childSize 
Result curRes = 
results) { 
r.sum; 
- r. size; 
new Result( root, childSum 
( res 
if 
null Il 
curRes.sum * res. size 
root. value, chi IdSize 
res. sum 
if 
return 
(cu rRes. size 
res curRes; 
curRes; 
work simulation (10 mins) 
just simple survey 
Good luck, fellows!

<https://www.jiuzhang.com/solution/subtree-with-maximum-average/#tag-highlight-lang-cpp>

05m 
IOS 
z 
o 
o 
I Test Cases I Output I 
Problem 
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.print/n() may 
not work in case of syntax/runtime error. The version of JDK being used is 1.8. 
Engineers at Amazon are interested in knowing which software component moves the fastest by counting the number of lines of code changed in that 
component and all subcomponents in the code base over a month. Component nodes have subcomponent nodes that make up the larger component. 
We've stored the components as a tree where each node represents one component that contains the number of lines changed that month. Software 
component speed is computed as the average number of lines of code changed for a given component and all its subcomponents. Engineers compute this 
software component speed for all such components which have one or more subcomponents. Find the component with the highest software component 
speed. Assume there will be at least one subcomponent in the tree and that there will be no ties. 
Input 
The input to the function/method consists of an argument: 
rootComponent, representing the root node. 
Output 
Return the reference to the component that has highest software component speed. 
Example 
Input: 
root Component: 
200 
120 
180 
110 2030 150 80 
Output: 
180 
Explanation: 
The component at the root of the tree has had 200 lines changed this month; its two dependencies have had 120 and 180 changed respectively. 
There are three components in this tree with the following changes: 
120 (110+20+30+120)/4 = 70 
180 (180+150+80)/3 = 130.67 
2 

<https://www.1point3acres.com/bbs/thread-475321-1-1.html>