

## 电面:

顺序统计树，找第  $K$  个节点。

- 1) 打印 000 到 123 所有的数，follow up，打印  $a$  到  $b$  所有的数（假设每一位都  $a \leq b$ ）
- 2) Next permutation
- 3) 栅栏  $N$  个木片，每一片可以刷两种颜色，相临三片不能同色，问几种刷法。

Q) Write a program to count the total number of pages reachable from a website.

For example, given "nytimes.com", count the number of pages reachable from there. More info on 1point3acres.com

You can assume you are given a function to fetch the page and extract the inner links, e.g.:

```
List<String> fetchPageAndExtractUrls(String url);
```

Q) Given a tiny computer with a 1 MHz CPU and 1 KiB of RAM memory

no input;

Only output is an LED light that says "I am done".

(1 MHz == 1 million instructions per second)

I load an arbitrary unknown program onto this computer.

How long do we have to wait in wall-clock time before we can prove the program has an infinite loop?

有个  $x, y$  坐标系的 array  $A[(1, 0), (0, 1)]$  都在一个圆上 圆心会给  $(0, 0)$

有第二个点的 array  $B[(8, 9), (12, -5)]$  求问 对于每个在  $B$  array 里的点，找出  $A$  里距离最近的点

第二题把  $A$  的点换成极坐标 sort 然后 binary search 最后没写完。

$N$  by  $M$  grids. red cells and black cells. Compute the # of squares without black cells. 就是 DP。所有 size 的 square 都算，比如一个  $2 \times 2$  的格子有 5 个 square， $3 \times 3$  的格子一共有 14 个 square。。。

跟 lc 那道 maximal square 有点类似。

一个无向图，不是全链接，节点上有数字。求最长连续递增数字串的长度。

9-1-2-3-4

| |

2-3

答案是 4: 1-2-3-4

1. 有一个 **tree**，不是 **binary**，define **weight as** 它的子数的点的个数。**weight** 不是 **treenode** 的 **field**
2. 手机 **app**，**response** 时间长，怎么处理。解决了怎么测试。不是写代码题。
3. 第一题的 **tree**，会常改变，要记录什么时间 **tree** 长什么样，用什么数据结构纪录。纪录很多很多有什么问题，怎么办。

一群人在排队，然后每个人知道自己前面有多少人比自己高，然后给你每个人的高度，要求 **reconstruct** 这个序列

给定一个二维数组，让你找里面有多少个 **3\*3** 的正方形，每行每列每个对角线的和都是一个给定的值。我这题想不到好方法，直接暴力遍历所有 **3\*3** 的方格去检查了

一个程序跑了很多次每次的结果不一样，问可能的原因

故意的随机

读写非法的内存.

**Race Conditions** 使用外部的變量，這個外部變量可能被别的程序修改

可能用到了共享的内存

**mamory leak**

可能依赖其他程序的输出

也許用到了外部的 **API**，調用失敗

3. 假设你在一个地铁站（地铁站有纸质的时刻表），手上只有一个 **stopwatch**，问如何确定现在的时间？然后写算法。。。。

就是有一堆 **player**，每个人 **beat** 其他人的概率已知。然后已知初始的对阵表，问给定一个 **player**，问他最后夺冠的概率是多少。

比如下面的例子

```

    ----
---          ----.
```

a--b c---d e---f g---h

我来讲下第五题，如果一共有  $N$  个选手， $N=2^h$ ，那么我们就需要  $h$  场比赛才能选出冠军。

用  $win[j]$  记录  $i$  对  $j$  的， $i$  赢的 win rate,  $win[j] + win[j] = 1$

如果有 0,1,2,3 四名选手 那么这个树里面，可能是这样的  $\{[0]\}$   $\{[0,3]\}$   $\{[0,1] [2,3]\}$  我们把最底层成为 0 层，root 称为  $h$  层。那么我们就是考虑  $root = 0$ ,  $root \rightarrow left = 0$ ,  $root \rightarrow left \rightarrow left = 0$ , ...  $root \rightarrow left * n = 0$  的可能性

我们现在要考虑的是从第  $K$  层的结果，推出  $k+1$  层的结果。

所以用 DP 去做。DP 大小为  $H*N = \log N * N$ ,  $DP[j]$  代表的是在第  $i$  层(round  $i$ ) 第  $j$  个选手没有被淘汰(win)的概率。由于 tree 的构造 你可以知道第  $i$  层中，活下来的最多为  $2^{(h-i)}$  个。

特别的  $i = 0$  的时候，  $dp[0][j] = 1$  for all  $j$ 。

我们最终是要求  $dp[h][0]$

好了，现在数学归纳法，假设我们已经求出了第  $k$  层，就是说我们知道了  $dp[k][0]$ ,

$dp[k][1] \dots dp[k][n-1]$

要用这些数据推出第  $k+1$  层。

OK 我们考虑  $dp[k+1][j]$  说明  $j$  在  $k$  层肯定没被淘汰，并且与  $j$  在  $k$  层的对局， $j$  赢了。那么这个概率就是

$dp[k+1][j] = dp[k][j] * (win[j][x] * dp[k][x])$  for all  $x$  might fight  $j$  in round  $k+1$ ,

注意这里的  $x$  不是由  $1 \sim N$  共  $N$  个 哦，很简单，比如考虑  $dp[2][3]$ ，那么 3 在第一轮已经赢了，说明 "2" 已经被打死了，所以不可能与 3 再对局。同样的  $dp[3][6]$  里面，和 6 对局的只可能是 0,1,2,3 而不可能是 4,5,7 (都已经死了)。所以 这个公式里求  $dp[k+1][j]$  需要 loop 的所有的  $x$  一共有连续的  $2^k$  个 (特别的，考虑下  $k+1=1$  和  $k+1=h$ )：那么 我们如何去求出  $xstart$  呢，

大家画一画图，看下构造就不难得出，如果第  $k+1$  层  $j$  是从 right child 胜出的，那么  $x$  在  $j$  的左边，

$xstart = j / (2^k) * 2^k - 2^k$ , 反之如果是从 left 胜出的，那么  $xstart = j / (2^k) * 2^k$

那么就可以写出我们的 loop 了：

input:

int n, h;

int win[n][n];

solution:

int dp[h][n] = {0};

for all j in 1-n:

dp[0][j] = 1;

for(int k = 0; k < h; k++){

//calculate dp[k+1]

for(int i = 0; i < n; i++){

int offset = i / (2^k) \* (2^k);

int fightWithLeft = (i / (2^k)) % 2;

int xstart = offset;

if(fightWithLeft){

xstart = offset - 2^k;

}

```

        for(int x = xstart; x < xstart + 2^k ; x++){
            dp[k+1] += win[x]*dp[k][x];
        }
        dp[k+1] *= dp[k];
    }
}
return dp[h][0];

```

那么这里  $Memory : O(h*N) = O(N\log N)$

$Time Complexity$ : 在第  $k+1$  层里, 对于每个  $i$ , 计算了  $2^k$  次, 所以是  $2^k * N$  次

那么一共  $(2^0 + 2^1 + \dots + 2^{(h-1)}) * N = (2^h - 1) * N = (N-1) * N$

所以是  $O(N^2)$  time

plus one (follow up: 10000 bits num)

```

implement sortedMap,
void put(key, value);
get(key);
getAt(index); //取得key排在index位置的value
不能用map, 只能用最基础的数据结构,

```

shuffle. 输入是  $[0, 2, \_, 3]$  输出是  $[0, \_, 2, 3]$ . 就是一个乱序数组, 其中缺少了一个值, 然后输出, 每个数值都在自己对应的index上面. 但是移动的时候, 只能把数字放在空缺的位置上, 要求移动的次数最少.

question 2, 现在同样给两个string, A和B, 但是B特别大, 可以想象成一个doc, 判断B中是否有sub-list 是A的permutation, 要求是consecutive sub list,

e.g. A= ["a", "b"], B=["c", "b", "a"], return yes. From 1point 3acres bbs  
["a", "b"], B=["c", "b", "d", "a"], return no.

已知一个data set , 包括学生名字 与 学生成绩 例如Michael 100 Lucy 80 分, 问 设计一个数据机构 满足两个操作 。1) GetNameByRank() 2) updateByName().

设计一个listener类, 能监听一些消息做一些处理

给一个tree, 找里面相同的subtree, 面完发现其实网上是有这道题的, 但我没怎么答出来, 在提示下说出来遍历+hash了但是没能写出代码来

UTF-8编码, 这个是之前面经里有的

给定一些点, 求能组成哪些三角形。给定一些边, 求能组成哪些三角形

给一个List<Float>, Float v, Float e, 让filter得到所有  $v - e \leq data \leq v + e$  的data  
这题的坑是不是 Float的 compare。不太确定的时候是不是应该google一下, 基础知识不过关啊

就是找连续数列, 在树里 (非二叉树), follow up 是一个图。

总之就是都用dfs走一遍

```
coding: give a list of string, define product as:
product(s1, s2) = 0 if s1 and s2 have character overlap
                s1.length * s2.length if s1 and s2 are distinct
```

find the max product of the list.

刚开始直接想到 string permutation,就说了sort和hashmap求frequency的两种办法,但阿三似乎都不满意,一直问能不能减少operation

实在想不出,厚颜无耻的问 can you give me a hint, recruiter很不错的提示说bitmap,眼前一亮呀!最后就是用bit operation 处理的

美国小伙。写 BigInt 类的你认为需要的接口函数,只写定义就好。然后让实现了 Constructor 和 Comp 函数(比较两个 BigInt)。让写了一大堆 test case。.

3. 同上美国小伙。一个 map{ "a":1, "b":2, "c":4}代表每个 string (或者 char, 无所谓)的 weight。现在有个函数 vector<string> genStrings(map m, int cnt)

让你返回一个 vector of string, 其中各个 string 出现的概率和之前给的 weight 成比例。比如给 cnt==700, 那么里面大概有 100 个 "a", 200 个 "b", 400 个 "c"。

问了好久好久的细节问题,比如 time complexity, space complexity, 还有 hash function collusion, 还有这个函数怎么测试。Random 函数是给的。

测试的话我就说用个大点的数 iterate 然后数结果。他问如果 "a" expect 出现的次数是 1M, 现在出现 1M+1K, 你觉得程序有没有 bug。我就说 normal distribution 有个 confidence interval 啊, 可以算出来然后看看啊。

有一个 tree, 不是 binary, define weight as 它的子数的点的个数。weight 不是 treenode 的 field

2. 手机 app, response 时间长, 怎么处理。解决了怎么测试。不是写代码题。3. 第一题的 tree, 会常改变, 要记录什么时间 tree 长什么样, 用什么数据结构纪录。纪录很多很多有什么问题, 怎么办。

#2:

<235, Obama> 代表在第 235 秒时, obama 得到了一个选票. 你现在有一个 list 的这种数据, 求在第 x 秒时谁的票数最多.

follow up: 你这个函数会被调用多次, 如何优化

## Find a sorted subsequence of size 3 in linear time

Given an array of n integers, find the 3 elements such that  $a[i] < a[j] < a[k]$  and  $i < j < k$  in  $O(n)$  time. If there are multiple such triplets, then print any one of them.

Examples:

Input: arr[] = {12, 11, 10, 5, 6, 2, 30}

Output: 5, 6, 30

Input: arr[] = {1, 2, 3, 4}

Output: 1, 2, 3 OR 1, 2, 4 OR 2, 3, 4

Input: arr[] = {4, 3, 2, 1}

Output: No such triplet

## Onsite:

1) 一堆 interval, 有叠加, 给一个值, 查询在不在这堆 interval 里 (会调用很多次)

Follow up, 给一个值, 查询多少个 interval 包含这个值 (会调用很多次)

(国人大哥面的, 可能会看到这个帖子, 非常感谢, 做题的时候给了很多引导。)

2) 一个有向图, 找出互相指向的点对数, (e.g. A 指向 B, B 指向 A, 算一对)

Follow up, 写一个类, 这个图会变化 (加点, 删点, 加边, 删边), 维护这样的点对数。

Follow up, 扯了扯大数据时候怎么分配到各个计算机上。

3) 论文演讲

4) 家族树, 每个点左右指针指向自己的父亲和母亲, 每个点存对应二叉堆的索引。

A) 给一个这种树, 给每个点标出对应二叉堆的索引值。

B) 任意给一个节点 (不需要输入根节点), 输出这个点所在的层数。

C) 任意给一个节点 (不需要输入根节点), 输出这个点和根节点的关系 (e.g. 是根节点父亲的母亲就输出“MF”)。

5) 两道 LC 原题 1. Anagram 2. Reverse bit

直接给出最优解还不断让优化, 优化涉及到系统设计。

Given a black and white image represented as a matrix, process the image in-place that only keeps the contour of the black region and turn all the inside pixels to white. 他强调 image 可能很大, 所以需要考虑怎样节约 memory

Design a game unblock me and implement the function to solve the game 就是把把木块移来移去, 最终使得红的木块能被移出出口~用 backtracking 来 solve 的~

Find a pair of Strings which don't have characters in common, and the product of the length is maximum

Find duplicate subtrees in a BT, and print out 这一轮脑子不转了，啊啊啊~面的不好，他提示我了好多~反正大的思路是有的，就是在每个 node 上找到 preorder 和 inorder traversal 的结果，然后跟 HashSet 里面已经有的比较，两个都有 duplicate 就代表是 duplicate subtree.

Skyline

一个 matrix，每个 matrix[i][j]有一个价格，给你一个 budget，问如何求出最大面积的子矩阵，使得子矩阵的价格之和不超过 budget

```
public int findMaxLen(int []arr,int budget, int []y){
    int maxLen=0;
    int sum=0;
    int start=0;
    for(int i=0;i<arr.length;i++){
        sum+=arr[i];
        if(sum<=budget){
            if(i-start+1>maxLen){
                maxLen=i-start+1;
                y[0]=start;
                y[1]=i;
            }
        }
        else{
            while(start<=i&&sum>budget)
                sum-=arr[start++];
        }
    }
    return maxLen;
}
```

//int [][]sub=new int[2][2]; top-left, right-bottom corners of the sub-matrix

```
public int findMaxRectangle(int [][]matrix, int budget, int [][]sub){
    int maxArea=0;
    int m=matrix.length;
    int n=matrix[0].length;
    int []tmp=new int[m];
```

```

int []y=new int[2];
for(int left=0;left<n;left++){
    Arrays.fill(tmp, 0);
    for(int right=left;right<n;right++){
        for(int i=0;i<m;i++){
            tmp[i]+=matrix[i][right];

            int maxLen = findMaxLen(tmp,budget,y);
            int area= (right-left+1)*maxLen;
            if(area>maxArea){
                maxArea=area;
                sub[0][0]=left;
                sub[0][1]=y[0];
                sub[1][0]=right;
                sub[1][1]=y[1];
            }
        }
    }
}
return maxArea;
}

```

, 给一堆长方形, 求它们的面积和。长方形自定义。

树形算术表达式 输出字符串形式, 在附加括号的处理上没处理好(后来一看 好简单, 心塞塞), 面试官等不及了就直接让 follow up 了, 输出数值, 先说把字符串输出 再用栈实现 果断被打断说就用刚才实现的函数改改做! 好吧就做了, 相对简单一点。

DFS:括号要判断当前节点是不是乘除的前提下 左子节点是不是加减 如果是的话 给左子返回来的表达式加括号 否则不用加 右子亦然

/\*

Are you familiar with memcpy()? Yes, C/C++ but mainly C

It takes a source, destination, and the size in bytes to copy.

```

void memcpy(void *dest, void *src, int size){
    char *srcc=(char*) src;

```



```
char *destc=(char*) dest;
```

//一开始写成了这样，因为好像以前在哪看过直接拷贝会有内存问题？

```
char *tmp=new char[size];
```

```
for(int i=0;i<size;i++)
```

```
    tmp[i]=srcc[i];
```

```
for(int i=0;i<size;i++)
```

```
    destc[i]=tmp[i];
```

//后来面试官说可以直接拷贝，就写成了这样

```
for(int i=0;i<size;i++)
```

```
    destc[i]=srcc[i];
```

```
}
```

memcpy is currently copying byte at a time. So if there are N bytes you have to make N iterations to the copy loop. Is there some way to improve this?

```
void memcpy(void *dest, void *src, int size){
```

```
    uint8_t *srci=(uint8_t) src;
```

```
    uint8_t *desti=(uint8_t) dest;
```

```
    int n=(size/sizeof(uint32_t));
```

```
    for(int i=0;i<n;i++){
```

```
        *((uint32_t*) desti) = *((uint32_t*) srci);
```

```
        desti+=sizeof(uint32_t);
```

```
        srci+=sizeof(uint32_t);. 1point3acres.com/bbs
```

```
    }
```

```
    //for the rest bytes
```

```
    n=(size%sizeof(uint32_t));
```

```
    for(int i=0;i<n;i++){
```

```
        *desti=*srci;
```

```
        desti++;
```

```
        srci++;
```

```
    }
```

```
}
```

// A Dynamic Programming based program to find minimum number

// insertions needed to make a string palindrome

```
#include <stdio.h>
```

```

#include <string.h>

// A utility function to find minimum of two integers
int min(int a, int b)
{ return a < b ? a : b; }

// A DP function to find minimum number of insertions
int findMinInsertionsDP(char str[], int n)
{
    // Create a table of size n*n. table[i][j] will store
    // minimum number of insertions needed to convert str[i..j]
    // to a palindrome.
    int table[n][n], l, h, gap;

    // Initialize all table entries as 0
    memset(table, 0, sizeof(table));

    // Fill the table
    for (gap = 1; gap < n; ++gap)
        for (l = 0, h = gap; h < n; ++l, ++h)
            table[l][h] = (str[l] == str[h])? table[l+1][h-1] :
                (min(table[l][h-1], table[l+1][h]) + 1);

    // Return minimum number of insertions for str[0..n-1]
    return table[0][n-1];
}

```

Q1: Assume you have a deck of cards. Each card has a number on it with no suit. We define "X of a kind" as X cards with same number on it ( $X \geq 2$ ). Determine if the deck can be fully divided into sets of "X of a kind".

Example: 3, 5, 3, 5, 3 -> True

3, 3, 5, 3, 3 -> False

Q2: Define "Straight" as 5 cards with consecutive numbers. Determine if the deck can be fully divided into sets of "Straight".

Example: 1, 2, 3, 4, 4, 5, 5, 6, 7, 8 -> True

Q3: Define "X-Straight" as X cards with consecutive numbers ( $X \geq 3$ ). Determine if the deck can be fully divided into sets of "X-Straight".

Example: 1, 2, 3, 4, 4, 5, 6 -> True

□□□:

```

bool dfs(const vector<int> &deck, map<int, int> counts) {
    if (counts.size() == 0) {
        return true;
    }
    if (counts.size() < 3) {
        return false;
    }

    int smallest = (counts.begin())->first;
    int count = 0;
    while (counts.find(smallest) != counts.end()) {
        if (counts[smallest] == 1) {
            counts.erase(smallest);
        } else {
            counts[smallest]--;
        }
        count++;
        smallest++;
        if (count >= 3 && (dfs(deck, counts) == true)) {
            return true;
        }
    }
    return false;
}

```

面的不好，特别紧张、先是嘘寒问暖，然后开始要做题了发现给的 google doc 链接是坏的，能进去，但是互相看不见对方。折腾了十分钟，开始问问题已经过去将近 20 分钟了。

问题是一个三位数的密码锁，假如初始密码比如是 000，你自己设定好的自己的密码比如是 777，有几个组合数当转动密码锁转到该组合会爆炸，给了一个 list，比方说[222,333,444,767]。然后问怎么转能转到自己的密码打开锁。答的不好，想了好几种最后用的 DFS，时间不够就让我 DFS 的过程。写着写着面试官一看没时间了说要我准备下一轮面试，特别不客气的挂了。我想死的心都有了

DFS 我觉得也可以，要考虑去重啥的，每个节点到下一个节点都有六个路径，走到 777 就算收敛，碰到要炸的就剪枝

大概就是，一个数组 1112223334445556677888....当中少 2 个数字，找到就行了。

```

public class FindTwoMissingElementsFromSortedArray {
    public static List<Integer> findMissing(int[] nums, int k) {
        List<Integer> result = new ArrayList<Integer>();
        if (nums == null || nums.length < 2) return result;
        helper(nums, 0, nums.length - 1, k, result);
        return result;
    }

    private static void helper(int[] nums, int start, int end, int k, List<Integer> result) {
        if (start > end) return;
        int leftMid = start + (end - start) / 2;
    }
}

```

```

int target = nums[leftMid];
while (leftMid > start && nums[leftMid - 1] == target) leftMid--;
int leftCount = leftMid - start;
if (leftCount == 0) {
    int rightMid = leftMid;
    while (rightMid < end && nums[rightMid + 1] == target) rightMid++;
    if (rightMid - leftMid + 1 == k - 1) {
        result.add(nums[leftMid]);
        helper(nums, rightMid + 1, end, k, result);
    }
}. 1point3acres.com/bbs
else if (leftCount % k == 0) {
    helper(nums, leftMid, end, k, result);
}
else if (leftCount % k == k - 1) {
    if (end - start + 1 == k - 1) { 鐳權□鑒宠□聰哄漣-涓€浜†竺錄喲並
        result.add(nums[start]);
        return;. more info on 1point3acres.com
    }
    helper(nums, start, leftMid - 1, k, result);
    helper(nums, leftMid, end, k, result);
}. 1point 3acres 聰哄漣
else if (leftCount % k == k - 2){
    if (end - start + 1 == 2 * (k - 1)) {
        result.add(nums[start]);
        result.add(nums[leftMid - 1]);
        return;
    }
    helper(nums, start, leftMid - 1, k, result);
}
}
}

```

```

public static void main(String[] args) {
    // int[] nums = {1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, 8};
    // int[] nums = {1,1,1,2,2,2,3,4,4};
    int[] nums = {1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 8, 8, 8};
    // int[] nums = {1,1,2,2,3,4,4,5,6,6};
    int k = 3;
    List<Integer> result = findMissing(nums, k);
    for (int d : result) {

```

```

        System.out.println(d);
    }
}
}

```

第一轮：。大概是个东欧小哥，先是聊聊自己对各种语言的看法，然后做题。发音有点不太清楚，题目交流花了一点时间。

给一堆 positive integer 如【10, 15, 20, 25】之类，然后可以对 integer 用上 cut 的方法(如果大于 10, 则一次 cut 产生一个 10), 看在有限的 cut 的次数下, 做多能 cut 多少个 10 出来。

可能描述的有点绕, 举个例子: 10 不需要 cut, 20, cut 一次就产生 2 个 10, 30, cut1 一次, 产生一个 10 和一个 20, cut2 次产生 3 个 10; 15 最多能被 cut1 次, 会产生一个 10。

**Greedy:** 现 cut 能被 10 整除的, if  $n / 10 = k$ , cut 能产生  $k - 1$  次能产生  $K$  个 10。

第二轮：一个俄罗斯大叔 1. 平面上一堆点, 判断是否关于某个垂直于 x 轴的线对称, 面经里有, 所以很快做完

**sort** 之后找 **median**, 因为既然关于这条轴对称, 那么 **median** 的值就应该是这条轴了。然后去掉所有在这条线上的点, 左右两边个数应该相同, 同时能两两对应

2. 一个含有 interger 的 matrix, 找出一个点使得左上角的 submatrix 和 到右下角的 matrix 的和相等

**DP:**  $dp[i][j]$  记录 (0,0) 到 (i,j) 的面积。

左上角的面积:  $dp[i][j]$  右下角的面积: 总面积 -  $dp[m - 1][j] - dp[i][n - 1] + dp[i][j]$ 。

第三轮：一个 mm, 刚来 google 三个月。

1. 面经上的 flower 题, 但没细看, matrix 中有 flower, statue, 还有 empty 的地方。statue 能挡住视野, 问站在哪个空的地方, 能看到最多的花 (只可以看上下左右, 四个方向)

**DP:** 每个方向分别对矩阵扫一遍, 比如计算向上看的方向:

$dp[i] = i \text{ is statue? } 0 : i \text{ is flower? } dp[i-1] + 1 : dp[i-1]$

2. 给一堆硬币, 可能有重复, 问可以组成多少种不同的面额, 没要求 coding

$2^n$  种可能, dfs

第四轮：国人小哥, 加一个 shadow, shadow 先进来, 以为是面试官, 所以对国人小哥表示下抱歉先问了简历

1. 简单 bfs 问题,

2. merge intervals

1. a) 给了个 board，每个 entry 有一个颜色，给定一个 entry 和一种颜色 A，把所有和该 entry 颜色相同的 connected board 涂上颜色 A。

### BFS

b) 给很多学生，纸笔，统计单词数。好像面经里出现过

2. 这轮崩了。。a) 给一个正整数 n，求能被 n 整除的且只由数字 1 组成的正整数的最小值，如果找不到返回 0，否则返回这个数的长度

b) 题大概是这样的，首先，一个二叉树，每条 edge 有 weight（想象成 graph），matching 代表了一组两两没有公共节点的 edges。现在让找一个 matching，满足两个条件，一是 cover 所有节点，二是 edge 之和最小。没做出来，提示下讨论了一种 recursion 的做法（因为是二叉树，所以可以左右找，总共要返回四种情况，实在水平有限，只能讲到这了）。。

**root = max(W(左边) + (DP(右子树) + DP(左孩子的左右子树)), W(右边) + (DP(左子树) + DP(右孩子的左右子树)))** 这样

3. topological sort 一堆有先后顺序要求的 Task

4. a) 不改变顺序的情况下，把 array 里的 0 都移到最后，leetcode 原题

b) 计算二叉树两个节点的最短距离。先找 LCA，再分别求高度。

Maximal square

Maximal rectangle

Skyline

给你一个无向图 G(V, E)还有一个正整数 k，G 表示 grapha，V 表示点，E 表示边，要求找到 G 的最的一个 subset G'使得 G'中每个点至少有 k 个 neighbor

先找到所有小于 k 个邻居的点，然后从 graph 里删掉相应的点跟他们所连的边，然后删的时候看跟他们相邻的点是否会小于 k 个邻居，如果是的话，就把相邻的点也给删了，一直删到所有的点都至少有 k 个邻居

第一轮，实现 iterator，给一个函数 a 返回 boolean, iterator 的 next 函数返回下一个满足函数 a 的元素。之后问了简历上的一个 project，他让我在白板上写写划划给他讲

第二轮，是有一个 stock feed 会返回股票时间戳和股票价格，然后因为会有错误，相同时间戳会返回多个股票价格（以最后一次接受的股票价格代表那个时间戳的价格），实现 得到股票价格的最大值和最小值还有股票价格的最新值，我用的 hashmap 和 priorityqueue。follow up 是如果 stock feed 出错率很低，如何改进。第二题是一个数组先递增后递减问如何得到 peak number，用 binary search 实现

第三轮实现 big Integer 的 add function，我只是用了一个很简单的 char 型数组实现的，不过楼主写的代码太长了，bug 也有点多，写的也不公整，面完整体觉得这轮面的最不好，面试官其实挺开朗挺好的，一直让我跟他说我的思路，但我能感觉到他觉得我跟他的交流比较少

第四轮是有一堆 query<ID, start\_time, end\_time>，按照 start\_time 排序，输出是<id, start/end, time>按照 time 排序，我最开始说全排，面试官说没有那么多的内存，后来我就改进了一下但还是用了 priorityqueue 实现，问了时间复杂度，然后问我如果是 multithread，每个 thread 处理的 query 是不 overlap 的，问时间复杂度是多少，我最开始没理解，最后也答错了

round 1:

黑白棋。给一个棋盘和一个棋子的坐标，判断这个棋子是不是活的。Leetcode 也有类似的比这个难得题目。DFS/BFS 看能不能走到棋盘的边缘就好。分析复杂度。

给一堆点，问怎么画凸包，说思路就好。这个面经上没见过，lz 当时是完全不知道凸包的概念。

round 2:

写有 weight 的随机数生成器，请参见以前的面经。写完之后问了如何测试。

**把概率累加，然后 Binary search**

给一个 int array，和一个这个 array 里面存在的数字，把这个数移到 array 的最后面。two pointers 就好。

round 3:

给一个 int array，找任意一个 popular number, popular number 就是出现次数大于等于 array length 的 1/4 的数。其实就是 Leetcode 169, 229 Majority Element. 第一问 unsorted array 用的 hashmap。第二问 sorted array 用的 binary search。lz 没有说 moore's voting algorithm 的做法感觉有点假。第二问复杂度，worst case 问的很细。

round 4:

类似 Leetcode 26, 80, remove adjacent duplicate elements from a list of characters.

类似 Lint code data stream median, 写个 API，有两个方法，addURL(String url) 和 getURL()，getURL()返回的是目前为止所有 URL 长度的中位数。lz 使用两个 heap 做的。follow-up: what if we know the range of the input，比如我们知道 URL 的大小不会超过 2k，那有没有别的 implement 的方法。这个没想出来请大家指教。

**用了一个长度为 2k 的数组，每个存对应长度 url 的 freq**

**凸包问题 可以看普林斯顿算法 I 的 5-6 节 有详尽解法**

(1) coding implement hash table (insert, find, remove)

(2) coding

you have 1 meter walkroad, and randomly generate rain, the rain is 1 cm. simulate how many rain drops to cover all the 1 meter  $[-0.01 \sim 1]$ .

merge intervals

(3) coding

give a dictionary with many words,

ape

apple

...

randomly generate a word from the dictionary.

calculate the probability from one letter to another and one letter to the end.

and the start probability of each letter

(4) coding

valid unicode (几个 bug, 自己修改了)

第一轮 印度小哥

题目是, 在发邮件的时候, 比如输入 ben, 下边会提示名字(FirstName, LastName)或者邮件以 ben 开头的人, 设计一个类来完成这个提示功能。假设每次我们返回最多 10 个这样的结果。

Follow up I, 如果希望返回的结果是 alphabetic 有序的, 比如输入 ben 的时候, benaa 在 benbd 前面, 怎么设计。

Follow up II, 如果我们希望 FN 是 ben 开头的在 LN 是 ben 开头的前边, 比如 ben Back 在 ben Smith 前面怎么办。

**Trie prefix tree**

第二轮

LeetCode 的 Zigzag Iterator

比如我们有一个 `Iterator<Iterator<Integer>>`, 这个里边是 iterator

i1 1, 2, 3

i2 4

i3 5, 6

然后结果返回 1,4,5,2,6,3

Follow up 是, 如果这些 iterator 都有 `hasPrevious()`, `previous()` 方法, 意思就是后退一步, 你的 class 也应该有这两个方法, 来后退一步

比如我们现在结果返回了 1,4



这时候原来的 iterator 变成.

i1 2, 3

i2

i3 5, 6

如果调用 previous(), 变成

i1 2, 3

i2 4

i3 5, 6.

最后姐姐跟我大概讨论了下 concurrent 怎么办 用个 lock, 或者用 singleton pattern, 对这个 synconize 这个 instance.

**记录每个 Iterator 调用了多少次 Next, 然后调用 previous 的时候找调用 next 次数最多的那个 Iterator.**

第三轮

上来先 warm up 了下, 如果我们要在 internet 传输数据的话, 我们要 compress 和 encrypt 数据。我们应该先 compress 还是先 encrypt

**先压缩后加密: 运算少、速度快**

如果你和另外两个朋友出去玩, 每个人付一部分钱, 比如你掏了 car rent, 另一个人付了 hotel 等等。最后回家了, 你们想 AA, 最后你们每个人付的钱都一样, 写个方法能返回谁应该给谁多少钱

比如三个人分别掏了 5, 3, 1, 那么 a[2]应该给 a[0] 2 刀

如果现在有 n 个人的话, 应该怎么办

写完了之后, 大哥问, 最多需要多少次 transaction, 一个人给另一个人钱的话算一次, 我心想的是  $O(n - 1)$  就是  $O(n)$ , 大哥说 specific answer, 就是  $n - 1$  次

然后 Follow up, 如果每次 transaction 特别麻烦, 不管是时间还是空间都特别麻烦。如果不用考虑 everything, 不考虑 cpu, 不考虑硬盘, 我们想让这个次数最少, 怎么办。

**Greedy** □□□□□□□, **brute force**□□□□□□□□

第四轮 一个亚裔, 还有个白哥哥旁听

题目大概是, 每次用户会调用一个方法 double next(double v) 然后函数返回的是这个数之前的 windowSize 个数的 average

比如 windowSize 是 3, call 了 next(10) next(11) next(3) call(1), 第一个返回 10, 第二个返回 10.5, 第三个返回 8, 第四个返回 5

因为我用了 Deque 来保存之前的数据, 我以为他会问精度的问题, 我记得面经里有人发过, 结果没问。。。所以 Follow up 是 如果不用现成的 Deque 这个 class, 你怎么办。好像用个链表更好写。。。我作死说可以用一个 rotated array 来模拟这个功能, 其实也挺简单

**Window Average: queue**

1. 给一个有序数组，按顺序输出数组的平方

直接两个 **pointer**，一个从头一个从尾部，比较绝对值大小

2. leetcode 原题，找出存在的只包含两个字符的最长字符串。

[Longest Substring with At Most Two Distinct Characters](#)

3. 迷宫问题，找路径，NP 问题。

**NP, P, NP hard**

4. find target in rotated array。

[Search in Rotated Sorted Array](#)

**Leetcode range sum query**

**Leetcode word abbreviation**

第一个是要求写 abber 的功能，follow up 是给一个很大的字典生成这个字典里面所有单词的缩写，里面不少字符串可能会有同样的缩写所以如果之前的缩写用过了就要改缩写方式，从短到长这样

举例就是：

international -> i11l i10al in10l 遇到相同缩写的单词就这么排下去处理重复

**leetcode word break 2,**

第一轮：

多叉树，每个节点是一个整数，求书中最长递增路径比如 5,6,7,8,9

看似简单其实有不少坑，感觉就是跪在这一轮了，所以后边面的反而很放松。。

第二轮：

四叉树压缩黑白图片，一个图片递归分成 2x2 四部分，如果一个区域颜色一样就设为叶子节点，算黑像素比例

**Quad Tree**

follow up 是给两个图片，把白色视为不透明，黑色视为透明，重叠在一起，返回一个图片，都用四叉树表示

第三轮：

一堆密码箱，每个密码都是四位 0-9 之间，算一个暴力破解序列，包含所有可能的四位序列，让这个序列尽量短

给了一个贪心算法，代码写的比较长，而且没 bug free

**Hamilton path?**

第四轮：

系统设计题，设计 google map 后端存储，这个面试官口音极重，交流无力，希望大牛指点一下 google map 后端是怎么做的，怎么 scale？

最后一轮，求  $\sum(n^i)$  就是  $1+n+n^2+n^3+\dots+n^N$ ，快速写了一个  $O(N)$  之后让我优化，其实这个二分  $O(\lg N)$  很容易想的，但是当时用了很长时间表现不太好。

#### Leetcode Power

第二题是个矩阵，每个节点是一个计算机，计算机之间传一个文件的 cost 是节点值  $\times$  路径长度，选择所有计算机为接收端，求所有文件传输的 cost

快速写了个暴力方法

尝试动态规划无果

后来想到可以 cache 所有行列的 cost 和，正这一边反着一遍，然后状态转移就是  $O(1)$  了，但是没时间写了，在他提议下写了一个一维特殊情况的代码，中间有个加号还忘了写，算是 sloppy coding 吧

1

#### lc 66, Plus One

2

1)

Unique Paths 变体

$m \times n$  的矩阵，每一格有一个数字

求从左上角到右下角，和最大的路径

followup：如果两个人一起从左上角走，每个人到了一个格子之后会把该格子的数清零。求两个人走的路径和的最大值

2)

给一个  $5 \times 5$  的矩阵，把数字 1 到 25 填进不同的格子，要求相同行相同列的数字递增

求有多少种放法。

3

给一个字符串  $s$  和整数  $k$

求包含  $k$  个不同字符的滑动窗口的最大长度

#### Longest Substring with At Most Two Distinct Characters

4

#### lc 212. Word Search II ( print )

写一个生日提示系统，每次查询时显示下一个谁过生日。这个有点偏 system design, 比如他想看到你注意时区啊啥的很多细节（并不知道他是不是在黑我）。我直接说算法了，二分搜索

判断一个数字是不是质数。要用到两个 trie。同时遍历两个 trie 是最优解。

Plus One (negative)

Merge Interval

第三轮是个印度小哥，带了个白人 shadow, 第一道给一个 two D garden , 每一个 slot 可以是 flower 或者 Wall. 找一个合适的位置，让游客可以看到最多的 flower. 可以站在 flower 上，不能站在墙上。如果被墙挡了，就看不到墙后面的花。然后游客只能竖直或者水瓶看，不能看对角线。。比如

[ f, x, x, w, f],

[f, f, x, x, x],

[x, x, f, w, f],

[f, f, x, w, x]]

这样，{3, 0} 和 {1,4} 都能看到四朵花。

valid graph tree.跟 LEETCODE 不同的是，这个图是有向

Node {

int id;

Set<Node> children;

}

public boolean isValid(List<Node> graph) {}

开始用拓扑排序做完,发现不太对。。因为可能两个 Parent 有相同的 Child，但是 Tree 中这样不合法。

拓扑排序，另外判断每个点的入度不能超过一

Add two number LRU

HTML DOM 设计结构 比较字符串 迭代器遍历

第一轮 韩国小哥，入职两年了，题目：NxN 的矩阵，给一个值 M, 求一个新的矩阵，新矩阵的每一个元素是原来矩阵 MxM 中的最小值

第二轮 被放鸽子，后来最后补了一轮，题目：1. data stream 怎么做 random sample；2. 给 N 个 office，以及每个月 office 的假期，还有 office 之间的 adjacent matrix 问该如何变换 office 才能最大化假期；

### Reservoir sampling

第四轮 中国大哥 加 shadow, 题目：1. search in rotated sorted array 2. edit distance

第五轮 白人小哥，估计是德国人，题目：给一个 string, 找到其中的数字，然后在前面填充给点数量的 0；follow up 是怎么排序这种 string 要求 aaaa9bc < aaaa19bc..

第一轮：

1. 实现一个 class, 支持 get(key), put(key, value), delete(key), getRandom()四个操作。用了 hashmap 跟 ArrayList，删除时候跟队尾 swap 一下达到 四个 O(1).
2. A="abcde" B="abcxde" 除了 x 其他字符都一样 找出那个 x

第二轮

1. longest substring at most K distinct char. 高频题，快慢指针+hashmap 计数。优化点做法就是用双向链表类似 LRU 的方法做，hashmap 不计数，只用来更新每个 char 最后的位置，更新时把结点拉到 LRU 的 tail 并删除 head。
2. 一个 list, 含有 k 个数，然后给 N，让等概率的从 1-N 里选出不在 list 里的数。这题一下子问懵了，纠结半天怎么算那个等概率。

第三轮：

午饭后整个人都不在状态，来了一道看起来很简单数组查询题，就是一个 sorted array, 若干位置是空的，给一个 target 求第一个比 target 小的 position. LZ 直接开写 binary search, 结果发现坑挺多，遇到空位置移动指针的时候怎么都有 bug, 最终还是没能写出能 work 的代码，这题看着简单写起来不容易，也许上来应该先写个 O(N)的理一下思路。这一轮一定被给了低分。 . Waral 錦氫□鏈爰湾澶氣构筠◆,

第四轮：

多叉树序列化反序列化，写的磕磕绊绊。。这道刷题刷漏了，结果偏偏考到，真是验证墨菲定律

第一轮, 先扯谈, 然后出题, 地里有, 设计一个 java 的 Object 的 serialization, 关键部分是如果 object 的 attribute 之间有 reference loop 怎么办? 比如 object H 里面有两个子 object, H1 和 H2, H1 里又有 H2, H2 里又有 H1, 问应该如何设计. 讨论了一下, 思路大概是给子 object 一个 reference id, 然后 H1 里存 H2 的 reference, 用 reference:["一个 id"]表示, 同样的在 H2 里存 H1 的 reference.-google 1point3acres

第二轮, 地里有类似的, 一个公司在每个国家都有办公室, 每个国家的节假日不同, 每个月最多飞去一个国家, 而且最多只有 6 个小时的行程, 给定国家的 list, 每个国家每月的假日数量, 每两个国家之间的飞行时间. 不过这次不求最大化节假日, 只要求 build graph. 讨论了下怎么设计 vertices, 思路是应该每个国家每月的组合作为 vertices, 而不是仅仅只是每个国家.

第三轮, 地里有, infinite input stream, 求 top 10 input. 很多方法, 可以维护一个 size 为 10 或者 11 的 window.

第四轮, 常见的 task dependencies 问题, A->B, C->D, B->D, 输出 task 安装顺序. topological sort, 参见 leetcode: Alien Dictionary.. 1point 3acres 璁哄漕

第五轮, 1. 一些地区不喜欢带 4 的楼层, 给定最高楼层 N, 求实际楼层数. 就是说楼层没有 4, 14, 24, 34, 40-49, 54, ... 我朋友一开始想用交集并集做, 把 4, 14, 24, 34 作为一列, 把 40-49 作为一列这样的思路, 后来发现好麻烦, 经过提醒, 直接 recursive 处理, 比如 221 层楼, 那么就是 2 个 100 层楼的情况+2 个 10 层楼的情况+1 楼, 然后问时间复杂度是多少

2. 判断 subarray 是否有相加等于 0 的, subarray size 不定. 记得大概是 leetcode 原题, 做法就是扫一遍 array 依次相加, 然后存到 HashSet, 如果出现相同数字, 那么意味着存在 subarray 相加等于 0. 比如 2, -5, 2, 3, 1. 那么扫一遍依次叠加之后是 2, -3, -1, 2, 1. 有两个 2, 意味着从第二个 2 到第一个 2 中间的 subarray 的和是 0. 就是说  $\text{Sum } i \text{ 到 } j = \text{Sum } 0 \text{ 到 } j - \text{Sum } 0 \text{ 到 } i$

第一题：给两个长度相等的 string，如果两个 string 对应位置的字母不相同记为一个 distance，如果现在能够交换一次其中一个 string 中任意位置的两个 char，返回能够将 distance 缩到最小的两个 char 的 index（如果有多个最优解只返回一个）；第二题：有 3n 个数围成一个环，取走其中一个的

话会顺带去掉这个数相邻的两个数（这两个不计入总和），剩下的继续围成环，问取走  $n$  个数构成总和的最大值。

Android 手机手势解锁的所有可能性（至少连接 4 个点，至多全部 9 个），需要考虑三点一线 invalid 的情况，如果考虑成 1-9 九宫格的话就是从 1 连到 3 这样的路径是不允许的。

Leetcode 的 serialize and deserialize binary tree 的变形，不需要存每个 node 的 value，只需要结构。

第一题：超大文件按行去重，如果有多行内容是一样的，去除其中重复行只保留一行；第二题：实现一个 size 很小的 hashtable（情景是：实际使用中用到的 entry 很少，如果正常 hashtable 会导致很大的 overhead，所以需要设计一个这种情景下 overhead 很小的 hashtable）

1. a) 给了个 board，每个 entry 有一个颜色，给定一个 entry 和一种颜色 A，把所有和该 entry 颜色相同的 connected board 涂上颜色 A。follow up: DFS 和 BFS 有什么优缺点，时间空间复杂度

b) 给很多学生，纸笔，统计单词数。好像面经里出现过

2. 这轮崩了。。a) 给一个正整数  $n$ ，求能被  $n$  整除的且只由数字 1 组成的正整数的最小值，如果找不到返回 0，否则返回这个数的长度

b) 题大概是这样的，首先，一个二叉树，每条 edge 有 weight（想象成 graph），matching 代表了一组两两没有公共节点的 edges。现在让找一个 matching，满足两个条件，一是 cover 所有节点，二是 edge 之和最小。没做出来，提示下讨论了一种 recursion 的做法（因为是二叉树，所以可以左右找，总共要返回四种情况，实在水平有限，只能讲到这了）。。。more info on 1point3acres.com

3. topological sort 一堆有先后顺序要求的 Task

4. a) 不改变顺序的情况下，把 array 里的 0 都移到最后，leetcode 原题

b) 计算二叉树两个节点的最短距离。先找 LCA，再分别求高度。

1

lc 66, 原题 Plus One

2

1)

Unique Paths 变体

$m * n$  的矩阵，每一格有一个数字

求从左上角到右下角，和最大的路径

followup：如果两个人一起从左上角走，每个人到了一个格子之后会把该格子的数清零。求两个人走的路径和的最大值

2)

给一个 5\*5 的矩阵，把数字 1 到 25 填进不同的格子，要求相同行相同列的数字递增。鐏櫳口聰哄  
漕-涓€浜€-涓€文塔鎗

求有多少种放法

```
int count(int n) {
    map<vector<int>, int> countMap;
    countMap[vector<int>(n, 0)] = 1;
    for (int i = 1; i <= n * n; i++) {
        map<vector<int>, int> newCountMap;
        for (auto cur : countMap) {
            vector<int> curVec = cur.first;
            int curCount = cur.second;
            for (int j = 0; j < n; j++) {
                if ((j == 0 || curVec[j] < curVec[j - 1]) && curVec[j] < n) {
                    curVec[j]++;
                    newCountMap[curVec] += curCount;
                    curVec[j]--;
                }
            }
        }
        countMap = newCountMap;
    }
    return countMap.begin()->second;
}
```

解释一下：这里是逐个填入 1 到 25 的，dp 是用在了“只考虑 count，不记录结果”这一点上面。其中 countMap 是一个把 vector 映到 int 的 map，vector 就表示当前的选取方法，而 int 是选取方法对应的 count 数，

比如 vector 是 [4,3,2,1,1] 就表示第一行填了前 4 个，第二行填了前三个这样这样。（因为  $4+3+2+1+1=11$ ，所以当前一定是在处理  $i=11$  这个数）每次一个大循环的 loop 结束后，newCount 里面就存了填入第 i 个数后，所有的填法以及每个填法对应的 count。所有可能的状态数大概是 5! 种（也就是非严格单减序列的个数，比 5! 稍多，不过差不太多，）每个状态数会用常数时间（严格来说是  $O(5)$ ）处理，所以整体的时间比最终的结果数小很多。

3

给一个字符串 s 和整数 k



求包含 k 个不同字符的滑动窗口的最大长度

4

lc 212. Word Search II ( print ) . From 1point 3acres bbs

一共四轮，有一轮半是工作经验，自我介绍，项目和 behavior。

```
double myPow(double x, int n) {  
    if(n==0) return 1;  
    if(n<0) {  
        n = -n;  
        x = 1/x;  
    }  
    double ans = 1;  
    while(n>0){  
        if(n&1) ans *= x;  
        x *= x;  
        n >>= 1;  
    }  
    return ans;  
}
```