

<http://www.1point3acres.com/bbs/thread-308322-1-1.html>

LC 43. Multiply Strings + 带时间戳的map: put(k, v, timestamp) get(k , timestamp)EX:

put(1, haha , 3)

put(1, ha, 5)

get(1, 0) -> null

get(1, 4) -> haha

get(1, 6) -> ha

Follow Up: ArrayList, LinkedList, BST, balancedBST 分别get & put 时间复杂度

两道比较经典的题，第二点解释的时候卡了一会，但是第二题上来就说主要是discuss不用写代码，但是其实剩的时间还挺多，不知道是好坏。其他都还好。希望是好消息。

```
class TimeMap{
public:
    TimeMap() {}
    bool set(int key, string val, int time) {
        timeMap[key][time] = val;
        return true;
    }
    string get(int key, int time) {
        if(timeMap.find(key) == timeMap.end()) {
            return "key not exist";
        }
        auto &keyMap = timeMap[key];
        auto it = keyMap.lower_bound(time);
        if(it == keyMap.end()) return keyMap.rbegin()->second;
        else {
            if(it == keyMap.begin()) {
                if(time == it->first) {
                    return keyMap.begin()->second;
                }
                else {
                    return "perhaps time too early";
                }
            }
            else {
                return it->second;
            }
        }
    }
private:
    unordered_map<int, map<int, string>> timeMap;
};
```

<http://www.1point3acres.com/bbs/thread-311343-1-1.html>

第一轮：给一棵树（不一定是二叉树）和一个节点（保证存在），按层次遍历顺序查询该节点的上一个节点，每层第一个节点的上一个节点规定为NULL。

Follow up1：经常查询的话如何优化。

Follow up2：给定父节点，如何支持插入操作

Time: O(N)

```
class Search {  
    void previousNodeOfLevel(TreeNode* root, TreeNode* target) {  
        queue<TreeNode*> q;  
        q.push(root);  
        while(!q.empty()) {  
            int n = q.size();  
            TreeNode* prev = nullptr;  
            while(n--) {  
                TreeNode* t = q.front(); q.pop();  
                //if(t == target) return prev;  
                m[t] = prev;  
                prev = t;  
                if(t->left) q.push(t->left);  
                if(t->right) q.push(t->right);  
            }  
        }  
        return nullptr;  
    }  
    TreeNode* queryNode(TreeNode* target) {  
        //Since the target is guaranteed in the tree.  
        return m[target];  
    }  
private:  
    unordered_map<TreeNode*, TreeNode*> m;  
};
```

第二轮：自我介绍，题是面经里的，给定两个单词，问第二个单词能否通过交换第一个单词中两个字母得到

Follow up：能否通过交换至多n次得到

ANS : <https://www.careercup.com/question?id=6464130037841920>

```
bool SwappableRec(string &s1, string &s2, unordered_map<char, unordered_set<int>> s2_char_to_swap_idxes, int idx = 0)  
{  
    for (int i = idx; i < s1.size(); ++i) {
```

```

//non-match char
if (s1[i] != s2[i]) {
    auto it = s2_char_to_swap_idxes.find(s1[i]);
    if (it != s2_char_to_swap_idxes.end()) {
        //all index of s1[i]
        vector<int> occurrence(it->second.begin(), it->second.end());
        for (int j : occurrence) {
            if (j > i && s2[i] == s1[j])
            {
                swap(s2[i], s2[j]);
                it->second.erase(j);
                if (SwappableRec(s1, s2, s2_char_to_swap_idxes, idx + 1)) {
                    return true;
                }
                it->second.insert(j);
                swap(s2[i], s2[j]);
            }
        }
    }
    return false;
}
return true;
}

bool Swappable(string s1, string s2)
{
    if (s1.size() != s2.size() ||
        s1 == s2)
    {
        return false;
    }
    unordered_map<char, unordered_set<int>> s2_char_to_swap_idxes;
    for (int i = 0; i < s1.size(); ++i) {
        if (s1[i] != s2[i]) {
            s2_char_to_swap_idxes[s2[i]].insert(i);
        }
    }
    return SwappableRec(s1, s2, s2_char_to_swap_idxes);
}

```

<http://www.1point3acres.com/bbs/thread-310605-1-1.html>

两点开始第一个，是一个native speaker，可能是我特地和hr说过希望和native speaker聊的原因。开始先问了我学历之类的，然后问了python里immutable和mutable的区别，然后问我第一篇论文.....然而我都记不清了，就聊了聊feature selection相关的问题。接下来开始做题，很简单的一道搜索题，题目是要解这个网址上的一个小游戏：

<http://www.mathsisfun.com/games/tilt-maze.html>。没有输入输出格式，全部自己想

LC 490

我本来想说可以dfs也可以bfs，但是他就提示我用bfs.....老子知道用bfs，这不是想告诉你思索过程吗摔！！！然后我就乖乖用了bfs，因为比较简单，所以说了很多细节，以及尽量将代码写的漂亮一些（比如先定义好接口之类的）完事儿后让我问问题，问了一下为毛用google doc写代码，这简直反人类呀.....

正式开始后，第一题是给你一个函数，可以返回指定区间内的随机整数，让你写一个返回三个整数的，还要求所有可能的three tuple概率一致。然后更改了题目要求，要求三者不同，我写了两个版本，一个while loop的，一个不需要while loop的

第二题是给定一些source 到 target的映射，要求找到那些target有三个source （原始题目没有提到source target之类的，是我给他们起的名字，反正意思差不多），把映射反过来就行。然后要求随机返回，恰好用到了第一题的函数因为光打电话打不通就浪费了好久，最后做完超了一点点时间，他还是给了我一个问问题的机会，我利用这段时陈述了我为什么去谷歌，以及问了一下他谷歌的生活之类的。然后就没了。

哎，不知道有木有希望进，希望人品++吧

<http://www.1point3acres.com/bbs/thread-310278-1-1.html>

第一轮是个印度小哥，问了两道LC median题。都刷过没啥好说的。第一题从一个array里sample k elements with equal probability. 直接给出in place解。 follow up: Reservoir Sampling (就是array 太大了放不进内存只能遍历一遍咋办) 都问了时间和空间复杂度。第二题：[LC 486. Predict the Winner](#)原题，给小哥讲了minmax算法的principal，用bruteforce解了一下，后续给了memoization优化解，感觉小哥很满意。

```
void selectKItems(int stream[], int n, int k)
{
    int i; // index for elements in stream[]

    // reservoir[] is the output array. Initialize it with
    // first k elements from stream[]
    int reservoir[k];
    for (i = 0; i < k; i++)
        reservoir[i] = stream[i];

    // Use a different seed value so that we don't get
    // same result each time we run this program
    srand(time(NULL));

    // Iterate from the (k+1)th element to nth element
    for (; i < n; i++)
    {
        // Pick a random index from 0 to i.
        int j = rand() % (i+1);

        // If the randomly picked index is smaller than k, then replace
        // the element present at the index with new element from stream
        if (j < k)
            reservoir[j] = stream[i];
```

```

    }

    printf("Following are k randomly selected items \n");
    printArray(reservoir, k);
}

```

第二轮是个白人大叔。因为之前第一场的时候答主手机出了点问题reschedule了，大叔好心安慰我说不要紧张，这不会影响面试feedback的啦，我当年毕业面试的时候blabla 10分钟(内心OS大叔 I'm good, 您快放题吧)。之后问了几个general question: 我看你简历上有Python和C++，那你来讲讲这两个语言的区别吧。然后又问什么样的project你会选择用Python，什么样的project你会选择C++。终于到coding环节了，题目是LC150. Evaluate Reverse Polish Notation升级版，运算要读入浮点数，除了加减乘除还有一个阶乘factorial。,

接下来高能来了，要求处理并抛出所有可能的异常！直接从算法题升级到工程能力的考察了QAQ. 楼主左思右想，在access stack的时候基本覆盖了所有可能的情况，加上实现了一个double factorial(double)，考虑了负数，input值域大于long long(其实大于INT\_MAX早就overflow了)，input不是整数(浮点精度问题)各种问题后，检查一遍感觉应该可以交卷了。结果大叔说：你的程序有bug！左思右想，我沉默两分钟说，抱歉我没找到...大叔直接定位到计算除法的那一行才想起来完全忘了division by zero了。只好尴尬的感谢大叔问了个小问题结束店面。

```

class Solution {
public:
    long long compute(stack<string>& operand, char op){
        long long op1 = 0L;
        long long op2 = 0L;
        if (operand.empty()){
            throw "Empty stack";
        } else {
            op1 = stoll(operand.top());
            operand.pop();
        }
        if (operand.empty()) {
            throw "Empty stack";
        } else {
            op2 = stoll(operand.top());
            operand.pop();
        }
        long long res = 0L;
        //cout << "op2 " << op2 << endl;
        //cout << "op1 " << op1 << endl;
        if (op == '+') {
            res = op1 + op2;
            if (res - op1 != op2) {
                throw "Overflow at +";
            }
        } else if (op == '-') {
            res = op2 - op1;
            if (res + op1 != op2) {
                throw "Underflow at -";
            }
        }
        return res;
    }
};

```

```

        }
    } else if(op == '*') {
        res = op1 * op2;
        if (op1 != 0 && op2 != 0){
            if (res / op1 != op2) {
                throw "Overflow at *";
            }
        }
    }

} else {
    if (op1 == 0L) throw "Divide by zero";
    res = op2 / op1;

    /*
    if (res * op1 != op2) {
        throw "Overflow at /";
    }
    */
}
return res;
}

int evalRPN(vector<string>& tokens) {
    stack<string> operand;

    for(string& str: tokens) {
        char c = str[0];
        if (str.size() == 1 && (c == '+' || c == '-' || c == '*' || c == '/')) {
            //operator
            try{
                long long res = compute(operand, c);
                //cout << to_string(res) << endl;
                operand.push(to_string(res));
            } catch(const char* msg) {
                cout << msg << endl;
            }
        } else {
            //operand(number)
            operand.push(str);
        }
    }
    return stoi(operand.top());
}
};


```

<http://www.1point3acres.com/bbs/thread-303542-1-1.html>

两phone interview

1. 给一组字符串，找出任意两个字符串有没有重复的字符，如果没有得到分数是两个字符串长度相乘，如果有得到0分，求最大的分数

### LC 318. Maximum Product of Word Lengths

2. 有很多房间，有些房间有钥匙，有一个房间有宝藏，求是否能从给定的房间出发找到宝藏

```
int strEncode(const string &s) {
    unordered_set<char> st;
    int res = 0;

    for(char c : s) {
        st.insert(c);
    }

    for(char c : st) {
        res += 1 << (c - 'a');
    }

    return res;
}

bool hasSameChar(const string &a, const string &b) {
    return (strEncode(a) & strEncode(b)) != 0;
}

struct Door
{
    Room* next_;
    int keyId_;
};

struct Room
{
    int keyId_ = -1;
    bool hasTreasure_ = false;
    vector<Door> doors_;

    Room(int keyId) : keyId_(keyId), hasTreasure_(false) { }
    Room(bool hasTreasure) : keyId_(-1), hasTreasure_(hasTreasure) { }
    void connect(Room *next, int keyId = -1) {
        doors_.push_back({ next, keyId });
        next->doors_.push_back({ this, keyId });
    }
};
bool hasPuzzleASolution(const Room* start)
{
    unordered_map<int, vector<const Room*>> lockedRooms; // keyid -> locked rooms
    unordered_set<int> keys{ { -1 } }; // collected keys
```

```

unordered_set<const Room*> explored; // rooms explored
stack<const Room*> s{ {start} }; // rooms exploreable

while (s.size() > 0) {
    auto current = s.top();
    s.pop();
    if (explored.count(current)) continue; // multiple paths could have led me here
    explored.insert(current);
    if (current->hasTreasure_) return true; // found a / the one treasure
    if (keys.count(current->keyId_) == 0) { // a new key
        keys.insert(current->keyId_);
        auto lrIt = lockedRooms.find(current->keyId_);
        if (lrIt != lockedRooms.end()) { // unlocked a room
            for(auto it = lrIt->second.begin(); it != lrIt->second.end(); ++it){
                s.push(*it);
            }
        }
        //for_each(lrIt->second.begin(), lrIt->second.end(),
        [&s](const Room* r) { s.push(r); });
        lrIt->second.clear();
    }
    for (auto door : current->doors_) {
        if (explored.count(door.next_) > 0) continue;
        if (keys.count(door.keyId_) > 0) { // already have the key
            s.push(door.next_);
        } else {
            lockedRooms[door.keyId_].push_back(door.next_);
        }
    }
}
return false;
}

```

<http://www.1point3acres.com/bbs/thread-310613-1-1.html>

下午刚刚结束了g家的背靠背两轮实习面试，下面是面经，求评分，求大米，谢谢！

一面：

1. 实现一个数据结构，能够实现存储interval（给的input默认是non overlap的），然后给一个整数，查找这个数在不在当前已有的intervals里面。

用了一个list存interval，每次insert的时候找到对应的位置，确保整个list是保持sorted的（按照interval的start）。每次查找数的时候，做一个binary search找到对应的interval，如果找不到就返回False。

follow up: 如何实现O(1)的查找？这个一开始我说在insert的时候用set存所有在interval里的值，然后查找的时候就是O(1)，不过就是insert操作的复杂度太高。stuck了快20分钟，面试官也没给什么hint，然后move to next question了。还求大神指点

2. 你有一堆coin，求从中拿出任意数量coin的unique sum。

这个和利口得subset题类似

Time: O( $2^{\text{total coin quantity}}$ )

```
unordered_set<int> coinSum(int i, vector<int>& coins) {
    if(i == coins.size()) return unordered_set<int>{0};
    auto next = coinSum(i + 1, coins);
    unordered_set<int> res;
    for(int n: next) {
        int cur = n + coins[i];
        if(res.count(cur) == 0) res.insert(cur);
        cur = n;
        if(res.count(cur) == 0) res.insert(cur);
    }
    return res;
}
```

With coin quantity

```
unordered_set<int> possible_sums(int i, vector<int>& coins, vector<int>& quantity) {
    if(i == int(coins.size())) return unordered_set<int>{0};
    auto remaining = possible_sums(i + 1, coins, quantity);
    unordered_set<int> ret_set;
    for(int q = 0; q <= quantity[i]; q++) {
        for(int next_sum : remaining) {
            int curr = next_sum + q * coins[i];
            if(ret_set.find(curr) == ret_set.end()) ret_set.insert(curr);
        }
    }
    return ret_set;
}
```

二面：

1.input是一个list，里面存的是按键的操作符，其中删除操作用“\_”表示，比如["a","b","\_","c"]表示的就是“ac”，“b”被删除了。给两个这样的list，判断其所表示的字符串是否是相同的。

先是定义了一个evaluate的function，input是一个list，用stack存字符，遇到“\_”就pop out上一个字符。对俩list分别调用，然后比对即可

```
bool backspace(string str, string pattern) {
    stack<char> st;
    for (char c: str) {
        if (c == '<' && !st.empty()) {
            st.pop();
        } else {
            st.push(c);
        }
    }
    for (int i = pattern.size() - 1 ; i >= 0; --i) {
        if (st.empty() || st.top() != pattern[i]) return false;
        st.pop();
    }
}
```

```

    }
    return st.empty()? true:false;
}
bool backspace2(string str, string pattern) {
    string res = "";
    for(char c: str) {
        if( c == '<'&& !str.empty()) res.pop_back();
        else res.push_back(c);
    }
    cout << res << endl;
    return res == pattern;
}
bool backspace3(string str, string pattern) {
    int i = str.size() - 1, j = pattern.size() - 1;
    int n = 0;
    for(; i >= 0; --i) {
        if(str[i] != '<' && n == 0){
            if(j >= 0){
                if(str[i] == pattern[j]) --j;
            } else {
                return false;
            }
        }
        if(str[i] == '<'){
            ++n;
        } else if(str[i] != '<' && n != 0) {
            --n;
        }
    }
    cout << j << endl;
    return j < 0;
}

```

follow up : 如何实现O ( 1 ) 的空间复杂度 ? 我用的两个指针 , 分别从俩list的右边开始 , 如果俩都不是"\_"就可以直接比较 , 如果是"\_" , 则继续往左统计"\_"的个数 , 然后再往左跳与"\_"个数相同的步数到达下一个比较的位置。最后如果俩指针都超过了最左端 (<0), 则二者相同

## 2.LC 731

<http://www.1point3acres.com/bbs/thread-310306-1-1.html>

给一个 seed\_list 比如{1,2,3}, 以及一个 expansion\_list 比如[{1,2},{2,4},{4,6}]。 seed\_list是一个set , expansion\_list是一组set , 要求在expansion\_list里找出所有与seed\_list有交集的set,求它们与seed\_list的并集。

follow up 1 : expansion\_list里与被选中的set有交集的set也要考虑进去。

follow up 2 : 去掉seed\_list , 将expansion\_list里有交集的都并到一起。

中间还有一些complexity分析 , 不要求优化 , 只要做出来就行,还是挺简单的...

//origin problem

```
vector<int> intersection1(vector<int>& seed_list, vector<vector<int>>& expansion_list) {
```

```

unordered_set<int> seeds(seed_list.begin(), seed_list.end());
unordered_set<int> res(seeds);

for(auto& expansion: expansion_list) {
    bool isIntersect = false;
    for(int i: expansion) {
        if(seeds.count(i) == 1) isIntersect = true, break;
    }
    if(isIntersect) {
        for(int i: expansion) {
            res.emplace(i);
        }
    }
}
return vector<int>(res.begin(), res.end());
}

//follow-up1
class DisjointSet{
public:
    DisjointSet(int N) {
        id = vector<int>(N, 0);
        for(int i = 0; i < N; ++i) id[i] = i;
        sz = vector<int>(N, 1);
    }
    int find(int p) {
        while(p != id[p]){
            id[p] = id[id[p]];
            p = id[p];
        }
        return p;
    }
    void connect(int p, int q) {
        int pid = find(p);
        int qid = find(q);
        if(pid != qid) {
            if(sz[pid] < sz[qid]) { id[pid] = qid, sz[qid] += sz[pid];}
            else {id[qid] = pid, sz[pid] += sz[qid];}
        }
    }
private:
    vector<int> id;
    vector<int> sz;
};

```

```

vector<int> followUp1(vector<int>& seed_list, vector<vector<int>>& expansion_list) {
    unordered_map<int, vector<int>> table;
    for(int i = 0; i < expansion_list.size(); ++i) {
        for(int value: expansion_list[i]) {
            table[value].emplace_back(i);
        }
    }
    int seed_idx = expansion_list.size();
    for(int value: seed_list) {
        table[value].emplace_back(seed_idx);
    }
    DisjointSet ds(seed_idx);

    for(auto& p: table) {
        auto& toConnect = p.second;
        int sz = toConnect.size();
        for(int i = 1; i < sz; ++i) {
            ds.connect(toConnect[i-1], toConnect[i]);
        }
    }
    unordered_set<int> res;
    for(int i = 0; i < expansion_list.size(); ++i) {
        if(ds.find(i) == ds.find(seed_idx - 1)){
            //connect set
            for(int j = 0; j < expansion_list[i].size(); ++j) {
                res.insert(expansion_list[i][j]);
            }
        }
    }
    return vector<int>(res.begin(), res.end());
}

```

//follow-2

//very similiar to 1, just need to find out how many root in the DisjoinSets

<http://www.1point3acres.com/bbs/thread-309867-1-1.html>

第一轮：

听不出口音应该是个老美，花了五六分钟让我介绍了下自己的经历和做过的项目什么的，没有针对项目问什么问题，直接开始做题。

一共两道题。第一题是给一个int的array，输出数字的最高出现频率。比如，input: [1, 2, 2, 2, 4, 4, 5], output: 3.

第二题是给一个int的array，其中只有一个数字在array中只出现一次，其他数字都是成对出现的（一个数字出现两次），输出这个单一的数字。比如，input: [1, 1, 2, 2, 3, 5, 5, 9, 9], output: 3

题不难，对复杂度有要求，(sorted)第二题要求优化到O(logn)。

O(N) XOR

O(logN) binary search

```
int findSingle(vector<int> nums) {
    int left = 0, right = nums.size() - 1;
    while(left + 1 < right) {
        int mid = (left + right) / 2;
        if(mid % 2 == 0) {
            if(nums[mid] == nums[mid - 1]) //check
                right = mid;
        }
        else {
            left = mid;
        }
    }
    else {
        if(nums[mid] == nums[mid - 1]) {
            left = mid;
        }
        else {
            right = mid;
        }
    }
}

if(nums[left] != nums[left - 1] && nums[left] != nums[left + 1]) return nums[left];
else return nums[right];
}
```

第二轮：

应该也是个老美吧，一上来直接做题。给了一道easy题，做完又把这题变了个形让重做。

原始题目是判断一个String内的括号是否有效(LC 678)。变形是说假设这个string内有多种不同的括号，不知道具体一共有多少种，判断这个string的括号是否有效。给了几个helper function可以直接拿来用，一个是可以检测是否是左括号，一个是检测是否是右括号，一个是输入左括号输出和这种括号match的右括号。

<http://www.1point3acres.com/bbs/thread-309876-1-1.html>

(Ph.D)

- 第一面

印度考官，人很有耐心，但是口音很重。

介绍自己做过的项目

判断一个矩阵是不是 Toeplitz Matrix  
(做了20分钟)

Follow-up:

如果这个矩阵很大，比如100M X 100M 怎么办

- 第二面

美国小哥，纽约分部  
寒暄，什么是Hash Table? (15 min)  
把任意数字改成7进制的 + test (15 min)  
comment: very good

- 加面

什么是hash table，什么时候会collision?

第一题，寻找一个binary tree中最多余的一条连接: 我用了hash table  
我开始说union find，她问什么是union find。。。然后我就换了hash table。感觉自己解释的不专业，previous node 什么的，肯定有专门的术语，非科班出身没办法

第二题，按权重生成随机数(在哪里见过?)

```
int sum_of_weight = 0;
for(int i=0; i<num_choices; i++) {
    sum_of_weight += choice_weight[i];
}
int rnd = random(sum_of_weight);
for(int i=0; i<num_choices; i++) {
    if(rnd < choice_weight[i])
        return i;
    rnd -= choice_weight[i];
}
```

还剩两分钟

<http://www.1point3acres.com/bbs/thread-309691-1-1.html>

第一次发帖，感觉跪了，发了一个simpson家谱的题目来，问我找两个node的edge数量，因为是家谱所以不是树形结构，这个问题用了common ancestor来解决了。

[img]blob:http://www.1point3acres.com/6db37b99-d1a5-4ee2-849a-f7f533ffc55c[/img]

类似这张图上：

bart到selma的edge距离是3，因为有三条边

bart到lisa距离是2，以此类推

//我的思路是用hash 和tuple做，hash用来记录深度信息，最后找到共同ancestor之后两个深度之和就是结果。。。

Thinking: single source shortest path

```
int twoNodeEdges(vector<vector<int>>& G, int start, int end) {
    queue<int> q;
    int dist = 0;
    q.insert(start);
    unordered_set<int> visited;
    while(!q.empty()) {
        dist++;
        int n = q.size();
        while(n--) {
            int t = q.front(); q.pop();
            visited.insert(t);
            for(int u = 0; u < G[t].size(); ++u) {
                if(visited.count(G[t][u]) == 0) {
                    if(G[t][u] == end) return dist;
                    q.push(G[t][u]);
                }
            }
        }
    }
    return -1;
}
```

<http://www.1point3acres.com/bbs/thread-305922-1-1.html>

(FT)

1. 问了简历上实习相关的东西
2. 哈皮number
3. binary tree in-order iterator， follow up：类似Java list iterator的previous() 和 hasPrevious()
4. 尴聊吹B，大叔说是Ad相关的组，他说他孩子可能都比我大。。。

<http://www.1point3acres.com/bbs/thread-307893-1-1.html>

(FT)

给一个同义词字典，然后给一个queries list，每一个query两个字符串，字符串里面用空格分隔为若干个字，要求输出每一个query是否是同义的

如同义词字典[["ab","cd"],["ef","gh"]]，表示ab和cd同义，ef和gh同义

queries: [["eeee ab cccc","eeee cd cccc"],["cd aaaa","ab aaaa"],["efs abcd","gh abcd"]]

应该返回[true,true,false] 每一个对应的字必须要相同或者是在字典里同义，这样才能返回true，第三个query里efs和gh不同且不同义所以返回false

Iz用的是Map<String,Set<String>>(一开始说了value也是string,然后小哥提醒一个字可能不止一个同义词，然后Iz就换set了)写完以后小哥说看着不错，然后就follow up了，也不知道有没有问题，问了一下复杂度，Iz回答的是query长度\*每个query里的字的长度

followup只是口述没有写代码，是存在同义转换，例如ab对应cd,cd对应ef,ef对应gh，则ab和gh也是同义  
Iz回答了bfs，然后问复杂度，Iz说bfs路径上所有字的同义词数量之和，然后小哥问如何优化，Iz先说了用个数组记录visited,如何小哥说还有啥优化，Iz想了一会说加memory,然后和他解释了一下memory工作原理。

<http://www.1point3acres.com/bbs/thread-309856-1-1.html>

(Strange... too many design)

第一轮：**LC 38. Count and Say**

还有一个设计题，时间不够了，没怎么写，大意就是给一个设计一个UI，在移动设备上支持显示行的可滚动列表：满足两个功能：1选择行和一个给定的偏移和水龙头的位置，2,调整某一行的高度

给一个已经有的scroll(int DeltaY)可以调用，实现 Row selectAt(int screenPosY) 和void resizeRow(RowIndex, int newHeight).

第二轮：两个设计题

第一，给你一些error type 和对应的 error code, 实现一个Enum可以返回不同的System error

第二，实现一个java singleton，写完之后他说要考虑一下死锁的问题，就加了一个lock

<http://www.1point3acres.com/bbs/thread-307988-1-1.html>

一共两题：

1. Binary Tree的三个遍历：先序，中序，后序

BST 用中序，然后遍历用两种办法写。recursion 和iteration

最后问了关于如何对比class的相同性

面试官是个印度人因此刚开始理解错意思了

2. Iterator

类似于LC 上的题。要自己设计一个iterator。要写next()和hasnext()函数

给你一个[2,4,0,9,5,3]

那输出就是[4,4,3,3,3,3]

奇数位是count偶数位是number

奇数位表示的是number会出现多少次

因此用

while(i.hasNext())

System.out.print(i.next())

输入是[2,4,0,9,5,3]结果就是[4,4,3,3,3,3]

Follow UP :

如果输入的是奇数长度的数组怎么办

```
vector<string> topCandidate(vector<pair<string, int>>& votes, int T) {
    unordered_map<string, int> m;
    vector<string> maxCandidate;
    int maxCount = 0;
    for(auto& p: votes) {
        if(p.second > T) continue;
        m[p.first]++;
        if(m[p.first] >= maxCount) {
            if(m[p.first] > maxCount) maxCandidate.clear();
            maxCount = m[p.first];
            maxCandidate.push_back(p.first);
        }
    }
    return maxCandidate;
}
```

<http://www.1point3acres.com/bbs/thread-309692-1-1.html>

map<time, candidates\_vote>

unordered\_map<candidates, map<time, vote>>

candi, timestamp

给一堆投票和相应的timestamp，输出小于给定的时间T得票最多的candidate

sorted or unsorted?

unsorted: O(N),

```
vector<string> topCandidate(vector<pair<string, int>>& votes, int T) {
    unordered_map<string, int> m;
    vector<string> maxCandidate;
    int maxCount = 0;
    for(auto& p: votes) {
        if(p.second > T) continue;
        m[p.first]++;
        if(m[p.first] >= maxCount) {
            if(m[p.first] > maxCount) maxCandidate.clear();
            maxCount = m[p.first];
            maxCandidate.push_back(p.first);
        }
    }
    return maxCandidate;
}
```

follow1: 再给一个长度k，输出小于给定时间T得票前k的candidate list

```

struct compare {
    bool operator()(pair<string, int> p1, pair<string, int> p2) {
        return p1.second > p2.second;
    }
};

vector<string> bestCandi (vector<pair<string, int>> votes, int time, int k) {
    unordered_map<string, int> voteBeforeT;

    for(auto vote : votes) {
        if(vote.second <= time) {
            voteBeforeT[vote.first] += 1;
        }
    }

    priority_queue<pair<string, int, vector<pair<string, int>>, compare> pq;

    for(auto candi : voteBeforeT) {
        if(pq.size() < k) pq.push(candi);
        else {
            if(candi.second > pq.top().second) {
                pq.pop();
                pq.push(candi);
            }
        }
    }

    vector<string> res;

    while(!pq.empty()) {
        res.push_back(pq.top().first);
        pq.pop();
    }

    return res;
}

```

follow2: 给一个candidate list , 求出最小的T使得使用follow1的function能得到这个candidate list

//traverse T, call follow-up2, compare

给定一个Synonym Pair List , 一个Query Pair List , 输出每个对query pair是否匹配 , query里的words顺序不变

follow1: synonym pair可以连通 , 比如pair (a, b) 和 (a, c) , 那么(a ,c)也是pair , 修改之前代码

follow2: query的words顺序doesn't matter , 判断query是否匹配

<http://www.1point3acres.com/bbs/thread-307197-1-1.html>

第一轮 , 给出坐标点 , 求四个点组成的长方形中 , 面积最小的 , 返回四个点坐标。

第二轮，输入一组string，求这些string合并成一个大string的所有组合。"abc" "def" "xyz" ->"abcdefxyz" "cdeabcxyz"类似这样的

```
//permutation
vector<string> permutationOfList(vector<string>& list) {
    vector<string> res;
    int n = list.size();
    if (n == 0) return res;
    vector<bool> visited(n, false);
    dfs("", 0, res, visited, list, n);
    return res;
}
void dfs(string temp, int len, vector<string>& res, vector<bool>& visited, vector<string>& list, int n) {
    if (len == n) {
        res.push_back(temp);
        return;
    } else {
        for (int i = 0; i < n; ++i) {
            if (!visited[i]) {
                visited[i] = true;
                dfs(temp + list[i], len + 1, res, visited, list, n);
                visited[i] = false;
            }
        }
    }
}
```

O

<http://www.1point3acres.com/bbs/thread-301567-1-1.html>

第一轮：阿三姐姐

问的Valid Parenthesis和print postorder from preorder and inorder. 第一题很快就秒了；第二题想了一会，最后在她的提醒下把算法写了出来，但是时间就到了。

第二轮：阿三哥哥

他的口音还是挺重的，要听得比较仔细才能听得懂。第一题问了LC 270. Closest Binary Search Tree Value，一开始想了一个O(N)的解法，后来在他的提醒下写了O(logN)。第二题问的是LC 334. Increasing Triplet Subsequence，这道题卡了比较久没有头绪，一开始的想法是brute-force O(N^3)，后来在他的提醒下缩短到了O(N^2)。回家之后一查觉得自己真的傻了，有很简单O(N)的解法。

<http://www.1point3acres.com/bbs/thread-302474-1-1.html>

两周前面的。

第一轮是一个美国大姐姐，上来说先转置一个矩阵（就是非常普通的转置，没有特殊要求，大概是warm up）。

用Python的list comprehension写了一个她好像没见过要求解释，于是一步步弄下来这个题也花了十多分钟（主要在给她解释）。

第二题是普通的拓扑排序，根据一系列subsequence要求还原原序列，需要检查没有解的情况。基于上一题的经验，从思路开始慢慢解释，先写了个特别暴力的，然后慢慢优化。稍微找找bug啥的。好像做完也就没有了。

第二轮是一个美国大叔，就一个题，经典的转汇率问题。随便写了个dfs，code也没咋说，花了不少时间在谈论需求啥的：如果有好多种结果怎么办，如果有矛盾的结果怎么处理，如果数据量特别大怎么办。基本上就靠嘴巴说过去了。最后剩了不少时间谈论人生。

<http://www.1point3acres.com/bbs/thread-304019-1-1.html>

第一轮第一题给一个收尾相接的（环形）linkedlist，让你隔一个删一个隔一个删一个；第二题是讲一个密室游戏，给你一堆房间，有些房间里面有钥匙，有些房间需要特定的钥匙打开，还有一个房间里面有宝藏，还会给你这些房间之间相连的关系，问你能不能从给定的某一个房间找到最后的宝藏，就返回0或1

```
struct Room {
    int id;
    int keyToOpen; // -1 for no need key
    vector<Room*> neighbors;
    bool hasTreasure;
    int hasKey;
};

class Solution {
    unordered_set<int> visited;
    unordered_set<int> keyHave;
    queue<Room*> needKey;
    bool foundTreasure;
    bool canFindTreasure(Room *room) {
        dfs(room);
        while(!needKey.empty()) {
            if(keyHave.find(needKey.front()->keyToOpen) != keyHave.end()) {
                dfs(needKey.front());
                needKey.pop();
            }
        }
        return foundTreasure;
    }

    void dfs(Room *room) {
        if(visited.find(room->id) != visited.end()) {
            return;
        }
        if(room->keyToOpen != -1 && keyHave.find(room->keyToOpen) == keyHave.end()) {
            needKey.push(room);
            return;
        }
        if(room->hasTreasure == true) {
            foundTreasure = true;
            return;
        }
        if(room->hasKey != -1) {
```

```

        keyHave.insert(room->hasKey);
    }
    for(auto neighbor : room->neighbors) {
        dfs(neighbor);
    }
}
};


```

第二轮第一题给你一个array，让你去掉其中重复的数，而且不需要in place；第二题利口原题，我居然没一下子想出来，就是给你一个先增后减的array，让你找到其中的最大值。

```

int findMax(vector<int> nums) {
    int left = 0, right = nums.size() - 1;

    while(left + 1 < right) {
        int mid = (left + right) / 2;
        if(nums[mid - 1] <= nums[mid] && nums[mid + 1] <= nums[mid]) {
            return nums[mid];
        }
        else if(nums[mid - 1] <= nums[mid] && nums[mid] <= nums[mid + 1]) {
            left = mid;
        }
        else {
            right = mid;
        }
    }

    if(nums[left] > nums[right]) return nums[left];
    else return nums[right];
}


```

<http://www.1point3acres.com/bbs/thread-303702-1-1.html>

第一轮只有一道题：给定若干个区间以及一个数，返回这个数对应的区间序号。例如区间为[1,10],[11,20],[21,30]，给了数字15，那么就应该返回1(对应的是第二个区间[11,20])。思路是用binary search，但需要同时判别左右两个边界。这道题我比较快给了思路，然后写出了代码。考官之后让设计几个test case测一下，然后发现存在一些问题，后来纠正过来了。之后问如果要scale up的话可以怎么做，我就说可以用多线程，然后把代码改了一下。后面问了要怎么处理corner cases。最后问还可以如何改进我写的代码，我没太明白他说的改进是什么意思，就问他加注释、统一代码风格这些算不算，然后他说可以。。。总之这个面试的题目很简单，但是不知道follow up答得算不算ok。

第二轮有两道题：第一题是给一个int，返回它的二进制的最高位1的Index。比如，对于int 4，二进制是100，最高位1的index就是2；对于int 1，二进制是1，最高位的index是0；对于0，返回-1；对于负数，因为最高位都是符号位，所以返回31。解法就是对于正数，不断的除2，然后计算除多少次能得到0，对于负数返回1，对于0返回-1。很简单的一道题。.

```

int highestIndex(int x) {
    if(x < 0) return 31;
    if(x == 0) return -1;
}


```

```

    for(int i = 31; i >= 0; --i){
        if((1 << i) & x) return i;
    }
    return -1;
}

```

第二题是leetcode 652 find duplicate subtrees。因为之前做过，所以比较快写出了答案。写完之后问了一下时间、空间复杂度，然后考官介绍了一下moutain view总部的情况，就结束了。

```

struct Subtree {
    int v;
    int lid; // id for left subtree
    int rid; // id for right subtree
    Subtree(int x, int l, int r) : v(x), lid(l), rid(r) {}
    bool operator==(const Subtree &r) const {
        return v==r.v && lid==r.lid && rid==r.rid;
    }
};

struct Hasher {
    size_t operator() (const Subtree& k) const{
        size_t key=k.v;
        return (key<<32)+(k.lid^k.rid);
    }
};

struct Comp {
    bool operator() (const Subtree& l, const Subtree& r) const {
        return l.v==r.v && l.lid==r.lid && l.rid==r.rid;
    }
};

```

<http://www.1point3acres.com/bbs/thread-303542-1-1.html>

1. 给一组字符串，找出任意两个字符串有没有重复的字符，如果没有得到分数是两个字符串长度相乘，如果有得到0分，求最大的分数

### 318. Maximum Product of Word Lengths

2. 有很多房间，有些房间有钥匙，有一个房间有宝藏，求是否能从给定的房间出发找到宝藏

<http://www.1point3acres.com/bbs/thread-305783-1-1.html>

第一轮：感觉美国口音，但不确定：找一个有向图里面reciprocated的edge的个数。

第二轮：国人大哥：找二叉树里面违反property的edge并删除。

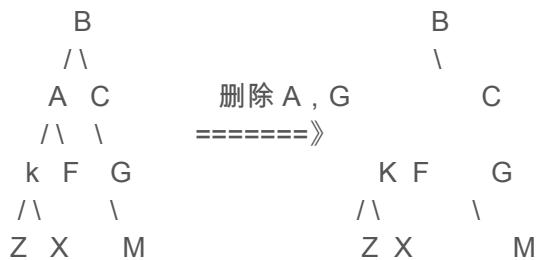
<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312747&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

今天下午面的 google intern 背靠背面经，发出来赞一下人品。第一轮 外国女生 Search组 话不多，语数很快。

第一题 warm up BinaryTree 求节点个数。

第二题 Binary Tree 删除其中的一些节点，返回所有剩下子树。

例如



```
class Solution {

public:
    vector<TreeNode*> deleteNode(TreeNode *root, unordered_set<TreeNode*> &deleted) {
        queue<TreeNode*> s;
        s.push(root);
        vector<TreeNode*> res;

        while(!s.empty()) {
            TreeNode *top = s.front();
            s.pop();
            if(top == nullptr) continue;
            helper(top, deleted, nullptr, s, true);
            res.push_back(top);
        }

        return res;
    }

    void helper(TreeNode* root, unordered_set<TreeNode*> &deleted, TreeNode *father, queue<TreeNode*> &s, bool left) {
        if(root == nullptr) return;
        if(deleted.find(root) != deleted.end()) {
            if(father != nullptr) {
                if(left) father->left = nullptr;
                else father->right = nullptr;
            }
            s.push(root->left);
            s.push(root->right);
        }
    }
}
```

```

    helper(root->left, deleted, root, s, true);
    helper(root->right, deleted, root, s, false);
}
};


```

### 第三题

给了一些字符，返回满足匹配的字符，注意大小写。

FooBar , FooBarTest , FooBarRed , FeeBar

Query:"FB" return all;

Query"FoB" return FooBar FooBarTest FooBarRed

第三题没有写完，讲了讲TrieTree的思路就结束了。

第二轮国人给的是lc437和经典knapsack dp题。

两轮的题目都不是很难，就是有点紧张，感觉讲的不是太好。

<http://www.1point3acres.com/bbs/thread-307902-1-1.html>

第一轮一个听口音一个白人小哥，问了一道感觉是原创题，不难。

There are 10 countries, each with a unique population. Your job is to create a random country generator that generates each country according to its population.

input形式自己选，我经过他的允许用的hashmap。

二十分钟差不多写完了，然后就开始聊天，小哥说 I have a very positive comment about you. 当时心里美滋滋。

```

class randomCountry {
private:
    int sum;
    vector<int> arr;
    randomCountry(vector<int>& populations) {
        arr = vector<int>(populations.size() + 1, 0);
        sum = 0;
        for (int i = 0; i < populations.size(); ++i) {
            sum+=populations[i];
            arr[i + 1] = arr[i] + populations[i];
        }
    }
    int getRandomCountry() {
        int num = rand() % sum;
        auto ptr = upper_bound(arr.begin(), arr.end(), num);
        return ptr - arr.begin();
    }
};

```

第二轮一个毛子大叔，姓氏貌似是卡斯帕罗夫，出了道dp的题，有点像leetcode frog，感觉也不难。

There is a ball on the nth stairs, and it wants to get to the ground(level 0). There are some sticky stairs, and once the ball lands on the sticky stair, it is stuck and can never move.

During odd turns, you can jump down 1, 2, or 4 stairs. During even turns, you can jump down 1, 3, 4 stairs.

Return the number of ways you can get to ground.

和面试官先确定了思路dp，写了个不用memoization的最好写解法。写的过程中我解释思路，然后面试官莫名其妙一直在催，还自言自语，为了听清他说话我还刻意留意慢点写，他先对code的方式抱怨了一大段，后面突然emmmmm了一会儿然后说it seems work， i like your way of writing it. Don't stop. 内心崩溃的。之后让我说时间复杂度，然后追着时间复杂的事情问了一大堆问题。我小心地问是否是希望我用memoization优化，他说是的，但是don't get ahead, let's focus on this。然后又花了十几分钟，最后只剩十五分钟给我优化，然后又开始催，写一行就说一行什么什么问题，然后要求我按照他的思路写，最后心态崩了，没写完，把思路说了一下，用一个 int[2][n]的array来存dp的subproblem的值。第二轮肯定是跪惨了。

不过结束之后反思一下，说到底确实自己实力不够好，不然dp的话应该可以秒出最简做法，不管客观情况如何

dp[i][0] : number of ways to go ground from stair i in odd round

dp[i][1] : number of ways to go ground from stair i in even round

dp[i][0] = dp[i - 1] (if i - 1 >= 0 && i - 1 is not sticky) + dp[i - 3] ... + dp[i - 4] ...

```
int stickyStairs (vector<bool>& sticky, int n) {
    vector<vector<int>> dp(n + 1, vector<int> (2, 0));
    dp[1][0] = 3;
    dp[1][1] = 3;
    for (int i = 2; i <= n; ++i) {
        if (sticky[i-1]) {
            dp[i][0] = 0;
            dp[i][1] = 0;
        } else {
            if (i - 1 >= 1) {
                dp[i][0] += dp[i-1][1];
                dp[i][1] += dp[i-1][0];
            }
            if (i - 2 >= 1) {
                dp[i][0] += dp[i-2][1];
            }
            if (i - 3 >= 1) {
                dp[i][1] += dp[i-3][0];
            }
            if (i - 4 >= 1) {
                dp[i][0] += dp[i-4][1];
                dp[i][1] += dp[i-4][0];
            }
        }
    }
    return dp[n][0] + dp[n][1];
}
```

<http://www.1point3acres.com/bbs/thread-307893-1-1.html>

面试的听口音像是美国小哥，也可能是口音不重的三哥，很干脆，直接上题，问题感觉不难。

给一个同义词字典，然后给一个queries list，每一个query两个字符串，字符串里面用空格分隔为若干个字

要求输出每一个query是否是同义的

如同义词字典`[["ab","cd"],["ef","gh"]]`，表示ab和cd同义，ef和gh同义

queries: `[["eeee ab cccc","eeee cd cccc"],["cd aaaa","ab aaaa"],["efs abcd","gh abcd"]]`

应该返回`[true,true,false]` 每一个对应的字必须要相同或者是在字典里同义，这样才能返回true，第三个query里efs和gh不同且不同义所以返回false

Iz用的是`Map<String,Set<String>>`(一开始说了value也是string,然后小哥提醒一个字可能不止一个同义词，然后Iz就换set了)写完以后小哥说看着不错，然后就follow up了，也不知道有没有问题，问了一下复杂度，Iz回答的是query长度\*每个query里的字的长度

followup只是口述没有写代码，是存在同义转换，例如ab对应cd,cd对应ef,ef对应gh，则ab和gh也是同义。

Iz回答了bfs，然后问复杂度，Iz说bfs路径上所有字的同义词数量之和，然后小哥问如何优化，Iz先说了用个数组记录`visited`,如何小哥说还有啥优化，Iz想了一会说加memory,然后和他解释了一下memory工作原理。

然后就是让我问问题，小哥好像挺愿意讲的，热情。

**LC 734 Sentence similarity**

**LC 737 Sentence similarity II (transitive)**

<http://www.1point3acres.com/bbs/thread-307885-1-1.html>

要求设计一个`readBuffer`的class，有`open`，`close`，`seek`，`read`的function，主要需要执行`read (void* buffer, size_t byte_to_read)`

前提是app每次都可能需要读同样部分的file，所以要cache过去已经读过的file

面试官还提出要求说为了便于`read from cache`，每次如果碰到没有cache过的file，直接就读入固定size的large data chunk ( e.g. 2GB )

follow up 问了下如果cache满了，该根据什么移除老的cache，来提供空间

不好意思，感觉自己写的不清楚，但实际上我也没真正理解题目，面试官复述了好几遍，导致最后时间不够了，所以code了没几行。。。

Read4K -> Read2G

<http://www.1point3acres.com/bbs/thread-307550-1-1.html>

昨天和狗家两名工程师们进行了热烈的谈话，内容如下：

1. 听口音像是是国人大哥，给了一个由0和1组成的矩阵，只可以走1且只能上下左右正向走，问有没有路线可以从第一行走到最后一行。.

例：

000100.

000100

001100

001000

001000

我已开始用了dfs，follow up是最小步数。写完后发现是 $O(n^3)$ ，大哥要求improve，于是尝试了dp，如果是1的话`dp[i][j] = dp[i][j-1] ++ || dp[i-1][j-1]++`，当时是号称是 $O(n^2)$ 但是发现没写对，输了。

说话声音一直很远，且有杂音，一句话可能得重复3遍才能听清，消耗了不少时间。结果我们还在讨论dp的时候下一个电话就来了，大哥说那算了就这样吧，留下了不work的dp ( 狗头 )

DFS- find path from first row to last row

```

struct hasher {
    size_t operator()(const pair<int, int> &p) const {
        return p.first ^ (p.second << 16);
    }
};

struct comp {
    bool operator()(const pair<int, int> p1,
                      const pair<int, int> p2) const {
        return (p1.first == p2.first) &&
               (p1.second == p2.second);
    }
};
bool findPath(vector<vector<int>> matrix) {
    if(matrix.size() == 0 || matrix[0].size() == 0) {
        return true;
    }

    unordered_set<pair<int, int>, hasher, comp> visited;
    stack<pair<int, int>> s;
    for(int i = 0; i < matrix[0].size(); i++) {
        if(matrix[0][i] == 1) s.push(make_pair(0, i));
    }
    while(!s.empty()) {
        pair<int, int> top = s.top();
        s.pop();
        if(visited.find(top) != visited.end()) {
            continue;
        }
        visited.insert(top);

        int row = top.first, col = top.second;
        if(row == matrix.size() - 1) {
            return true;
        }

        if(col - 1 >= 0 && matrix[row][col - 1] == 1) {
            s.push(make_pair(row, col - 1));
        }
        if(col + 1 < matrix[0].size() &&
            matrix[row][col + 1] == 1) {
            s.push(make_pair(row, col + 1));
        }
        if(matrix[row + 1][col] == 1) {
            s.push(make_pair(row + 1, col));
        }
    }
    return false;
}

```

BFS-find shortest path from first row to last row

Time: O( (MN)<sup>2</sup> )

```
int findPath(vector<vector<int>> matrix) {
    if(matrix.size() == 0 || matrix[0].size() == 0) {
        return true;
    }

    unordered_set<pair<int, int>, hasher, comp> visited;
    queue<pair<pair<int, int>, int>> s;
    for(int i = 0; i < matrix[0].size(); i++) {
        if(matrix[0][i] == 1) s.push(make_pair(make_pair(0, i), 1));
    }
    while(!s.empty()) {
        pair<pair<int, int>, int> top_pair = s.front();
        s.pop();

        pair<int, int> top = top_pair.first;
        int count = top_pair.second;

        if(visited.find(top) != visited.end()) {
            continue;
        }
        visited.insert(top);

        int row = top.first, col = top.second;
        if(row == matrix.size() - 1) {
            return count;
        }

        if(col - 1 >= 0 && matrix[row][col - 1] == 1) {
            s.push(make_pair(make_pair(row, col - 1), count + 1));
        }
        if(col + 1 < matrix[0].size() &&
           matrix[row][col + 1] == 1) {
            s.push(make_pair(make_pair(row, col + 1), count + 1));
        }
        if(matrix[row + 1][col] == 1) {
            s.push(make_pair(make_pair(row + 1, col), count + 1));
        }
    }
    return -1;
}
```

2.

- 给出勤表判断有没有连续三次迟到或者旷课一次以上 bool check(string att)
- HashTable和BST的优缺点和implementation
- 这个很有意思，你在debug的时候发现code有error，于是你添加了print语句发现bug没了，问怎么回事
- 如何test函数divide ( a,b )
- 第五个忘了。。。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=305745&fromuid=225868>

第一个是个亚裔小哥的感觉，问了一题目是给一串数组代表坐标，求问这些坐标中能连成的长方形中，面积最大的那个是多大，做这题时候脑子有点方，想了很久磕磕巴巴做出来也不知道他觉得怎么样。

naive: O( $N^4$ ) check any 4 points, if they are rectangle, compare its area to current maximum.

improve: O( $N^3$ ) traverse all points and save into a set, then choose any 3 points, check if the 4th one is in set.

第二个是个口音感觉是个白人男，问了题目比较简单，一共两题，这一轮感觉没啥问题。

第一题是给个序列，问最长连续递增子序列的长度 LC 300。

第二题是一种二叉树，但是每个树都只有parent信息，然后给两个node，问怎么找到共同祖先，一开始我用了set，然后他就问我怎么不用extra space做，他给了点提示，然后做出来了。

Thinking: LC160

```
TreeNode* findLCA(TreeNode* a, TreeNode* b) {  
    if (!a || !b) return nullptr;  
    TreeNode* pA = a;  
    TreeNode* pB = b;  
    while (pA != pB) {  
        pA = pA==nullptr ? b: pA->parent;  
        pB = pB==nullptr ? a: pB->parent;  
    }  
    return pA;  
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=304671&fromuid=225868>

第一轮：ABC面的

给一些关键词，为了防止老印，我把关键词换为中文吧，大家自行换英语单词

关键词 比如有 周杰伦，汪峰，王菲 三个

给一个句子 我是周杰伦和王菲的粉丝

返回 周杰伦，王菲

就是返回有什么关键词啦

follow up

关键词 有alias 比如，周杰伦（周董），王菲（王靖雯），etc

句子 我是周董和王菲的粉丝

返回 周杰伦 王菲

返回本人的大名不是绰号

第二轮：

老印

类似 LC329 非常相似 准备了肯定没问题

题不难，但是问巨多follow up，没答好，大家准备的时候请多思考edge cases，

其实follow up不难想，只要做完一道题后自己多思考一些衍生的东西

比如第二轮这道题除了返回最大的数值，怎么着能把路径也给返回了

这样的follow up很好想到吧，他也确实问了

Follow up: print longest path

```
class Solution {
public:
    int longestIncreasingPath(vector<vector<int>>& matrix) {
        int m = matrix.size();
        if (m == 0) return 0;
        int n = matrix[0].size();
        //LIP len to [i, j]
        vector<vector<int>> mem(m, vector<int>(n, 0));
        vector<vector<vector<int>>> path(m, vector<vector<int>>(n, vector<int>()));
        pair<int, int> maxPoint{0,0};
        int maxlen = 0;
        for (int i = 0; i < m; ++i) {
            for (int j = 0; j < n; ++j) {
                int len = dfs(i, j, m, n, mem, matrix, path);
                if (len > maxlen){
                    maxlen = len;
                    maxPoint.first = i;
                    maxPoint.second = j;
                }
            }
        }
        vector<int> longestPath = path[maxPoint.first][maxPoint.second];
        for (int i = longestPath.size() - 1; i >=0 ; --i) {
```

```

        printf("%d ", longestPath[i]);
    }

    return maxlen;
}

int dfs(int x, int y, int m, int n, vector<vector<int>>& mem, vector<vector<int>>& matrix, vector<vector<vector<int>>& path) {
    if (mem[x][y] != 0) return mem[x][y];
    int maxlen = 1;
    vector<int> maxPath;
    maxPath.push_back(matrix[x][y]);
    vector<int> d{0,1,0,-1,0};
    for (int i = 0; i < 4; ++i) {
        int dx = x + d[i];
        int dy = y + d[i+1];
        if (dx < 0 || dx >= m || dy < 0 || dy >= n || matrix[dx][dy] <= matrix[x][y] ) continue;
        int len = 1 + dfs(dx, dy, m, n, mem, matrix, path);
        vector<int> tempPath(path[dx][dy].begin(), path[dx][dy].end());
        tempPath.push_back(matrix[x][y]);
        if (len > maxlen){
            maxlen = len;
            maxPath = vector<int>(tempPath.begin(), tempPath.end());
        }
    }
    mem[x][y] = maxlen;
    path[x][y] = maxPath;
}

return maxlen;
}
};


```

<http://www.1point3acres.com/bbs/thread-307407-1-1.html>

两轮

第一轮：输入的string，有backspace

比如 abcde<< 就是 abc

比如 abcd<e< 也是abc

写个function测试两个是不是一样

这个比较简单

```

bool backspace2(string str, string pattern) {
    int i = 0;
    string res = "";
    for(int j = 0; j < str.size(); ++j) {
        if(str[j] != '<' && !str.empty()) res.push_back(str[j]);
        else res.pop_back();
    }
    return res == pattern;
}


```

```

}

bool backspace(string str, string pattern) {
    stack<char> st;
    for (char c: str) {
        if (c == '<' && !st.empty()) {
            st.pop();
        } else {
            st.push(c);
        }
    }
    for (int i = pattern.size() - 1 ; i >= 0; --i) {
        if (st.empty() || st.top() != pattern[i]) return false;
        st.pop();
    }
    return st.empty()? true:false;
}

```

第二轮：一个城市的公交车站 比如公车1号到abcd站，公车2号到cefg站。那么a-d就是1，因为只要坐一辆车，a-g就是2，因为要从c那边1号换2号

然后问 给了这个信息 要求出一个站到另一个站的最少换乘

(感觉就很现实googlemap哈哈哈)

用bfs可以 但是之前的公交车站的表示我写的有点乱 所有data structure要自己想

```

struct Bus {
    int id;
    unordered_set<char> stops;

    bool operator==(const Bus &a) const {
        return a.id == this->id;
    }

    Bus(int id, string stops) {
        this->id = id;
        for(char c : stops) {
            this->stops.insert(c);
        }
    }
};

bool hasIntersection(Bus a, Bus b) {
    for(auto stop : a.stops) {
        if(b.stops.find(stop) != b.stops.end()) {
            return true;
        }
    }
    return false;
}

```

```

int leastSwitch(vector<Bus> buses, char start, char target) {
    queue<pair<Bus, int>> q;
    unordered_set<int> visited;
    for(int i = 0; i < buses.size(); i++) {
        if(buses[i].stops.find(start) != buses[i].stops.end())
        {
            q.push(make_pair(buses[i], 0));
        }
    }
    while(!q.empty()) {
        Bus curr = q.front().first;
        int switches = q.front().second;
        q.pop();

        if(visited.find(curr.id) != visited.end())
        {
            continue;
        }
        visited.insert(curr.id);
        if(curr.stops.find(target) != curr.stops.end())
        {
            return switches;
        }
        for(int i = 0; i < buses.size(); i++) {
            if(curr == buses[i]) continue;
            if(hasIntersection(curr, buses[i]) &&
                visited.find(buses[i].id) == visited.end())
            {
                q.push(make_pair(buses[i], switches + 1));
            }
        }
    }
    return -1;
}

```

<http://www.1point3acres.com/bbs/thread-307252-1-1.html>

第一题，LC 734，Follow up LC 737，要求预处理。这轮的面试官大哥嫌弃我的烂口语，直接用中文和我面试的。。。不知道会不会对结果有影响。。。

## 第二题，1. merge intervals with value

PD: there are a series of continuous intervals, and each interval has a value. Initially, the  $i$ th interval is  $(i-1, i)$ , merge these intervals and return the result.

Merge Rule:

- We can only merge the  $i$ th interval with  $i-1$ th or  $i+1$ th interval. The value of new interval is the mean of these two original intervals.
- Define cost as absolute difference of two neighboring intervals,

every time merge two intervals with the smallest cost.  
c. If the smallest cost exceeds a threshold t, then stop.  
d. There may be multiple valid result, just return one.

for example:

value: 3 7 6 5 1

intervals: (0,0) (1,1) (2,2) (3,3) (4,4)

threshold: t = 2

first iteration:

min cost = |7 - 6| = 1, notice that |6 - 5| is also okay.

after merging:

value: 3 6.5 5 1

intervals: (0,0) (1,2) (3,3) (4,4)

second iteration:

min cost = |6.5 - 5| = 1.5

after merging:

value: 3 5.75 1

intervals: (0,0) (1,3) (4,4)

second iteration:

min cost = |3 - 5.75| = 2.75 > t = 2, stop

return [(0,0), (1,3), (4,4)]

Thinking: :linked list or map

```
class ListNode {  
public:  
    ListNode(int s, int e, double val) : start(s), end(e), val(val){  
        next = nullptr;  
    }  
private:  
    ListNode* next;  
    int start;  
    int end;  
    double val;  
}
```

```
vector<pair<int, int>> mergeList (ListNode* head, int t) {  
    double mindiff = 1e9;  
    ListNode* cur, mergeHead;  
    while(true) {  
        cur = head;  
        mindiff = 1e9;  
        if(!cur -> next) break;  
        while(cur && cur->next) {
```

```

        double diff = abs(cur->val - cur->next->val);
        if(diff < mindiff) {
            mindiff = diff;
            mergeHead = cur;
        }
        cur = cur->next;
    }
    if(mindiff > t) break;
    //merge
    mergeHead->val = (mergeHead->val + mergeHead->next->val) / 2;
    mergeHead->start = min(mergeHead->start, mergeHead->next->start);
    mergeHead->end = max(mergeHead->end, mergeHead->next->end);
    mergeHead->next = mergeHead->next->next;
}
}

```

<http://www.1point3acres.com/bbs/thread-296120-1-1.html>

剛做完了兩輪phone interview 面2018 summer的實習，來發面經求大米順便積人品-google  
Round 1 三哥

1. 自我介紹 + 作過啥project
2. 紿一個String 用outbound spiral方式輸出 (ex. abcde -> cdbae)

```

void spiral(string str) {
    int n = str.size();
    int mid = n / 2;
    string res = "";
    if(n % 2 == 1) res += str[mid];
    for (int i = n/2 - 1; i >=0; --i) {
        res += str[n-1-i];
        res += str[i];
    }
    cout << res << endl;
}

```

3. 紉一個String 找出最長的包含k個不同字母的substring **LC340**

Round 2 老美

1. 嘞賽 + 聊履歷
2. Regular Expression Match **LC 10**

這輪估計是掛了.....這題leetcode寫過但沒用心理解，只好從很naive的方法來想，想來想去想不出個好方法，最後在老美提示下用recursion作完，但時間也到了

<http://www.1point3acres.com/bbs/thread-307083-1-1.html>

第一题简单的不得了不说了，小学生水平.

## 第二题检查两个区间是不是有重叠

第三题给一个点和一堆一维区间，返回这个点是不是在区间里，分别用线性时间，对数时间和常数时间三种算法做

PS：忘了说了给的区间和点都是整数不会有浮点数

建一个一维数组过一遍所有区间，在区间里的数都设为真就好了

那么判断点直接查询数组，常数时间

问了大叔只要求query的时候是constant time就好了

不知道这是好事还是坏事，因为听说要送 hiring committee 再审查一次面试表现之类的  
人家一看我的题这么简单都和别人没有可比性。。。呵呵

<http://www.1point3acres.com/bbs/thread-304425-1-1.html>

## 第一轮

1. 给一个字符串a和一个forbidden character b，从字符串中移除所有的这个forbidden character.

样例：

"aabbaac", 'a' -> 得到 "bbc"

"Abcde", 'x' -> 得到 "Abcde"

面试官先让我写了很简单的一次性遍历的代码，然后问我能不能做到in place。我用两个变量pos1和pos2分别记录原字符串中当前字符和待生成字符串的当前字符位置(始终有pos2<=pos1)，还是一次循环。

Similar to move zero

2. 给出一个按字典序排好序的字符串数组dictionary，找出其中第一个不是以指定字符串prefix作为前缀的字符串。

样例：

[aa, aaa, ax, b, c], aa -> ax

[b, c, d], aaa -> b

```
int findfirstUnmatchPrefix(string prefix, vector<string> dict) {
    if(dict.empty()) return -1;
    int plen = prefix.size(), n = dict.size();
    if(dict[0].substr(0,plen) != prefix) return 0;
    int l = 0, r = n - 1;
    while(l < r) {
        int mid = l + (r - l) / 2;
        if(dict[mid].substr(0,plen) == prefix) l = mid + 1;
        else r = mid;
    }
    if(dict[l].substr(0,plen) == prefix) return -1;
    return l;
}
```

我先说可以一重循环扫一遍，每个字符串判断是不是以prefix作为前缀，复杂度是O(nk)，n是字典中字符串个数，k是字符串最大长度。面试官问我能不能优化，然后我苦想半天没想出来，他提示说可以试试binary search，我这才想到，用二分查找的方法去找字典中第left个到第right个字符里第一个不以prefix作前缀的字符串（初始时left=0,right=dict.size()-1），mid=(left+right)/2，先判断dict[left]是不是以prefix作前缀，如果不是直接返回dict[left]，如果是的话再判断dict[mid]，如果也

是的话那么按照字典序从dict[left]到dict[mid]全都是以prefix作前缀的，所以只需要判断dict[mid+1]到dict[right];如果dict[mid]不是的话那么就从dict[left+1]到dict[mid]中接着找。

## 第二轮

### LC 281. Zigzag Iterator

给出一个Flattener class, 它存储着一个嵌套的vector, 构造函数里传入的参数是vector<vector<int>>, 要求实现next()方法和hasnext()方法，其中逐次调用next()方法会依次取出每个vector的第一个元素，然后是每个vector的第二个元素，直到各vector被取空。hasnext()判断是不是所有的元素都取完了。

样例：

如果传入的是 {{1, 2, 3}, {4}, {5, 6}}, 那么每次调用next()依次取出 1, 4, 5, 2, 6, 3.

我先用一个vector<int> record\_vec 记录每个vector中取到第几个元素了，另外用一个变量pos记录当前遍历到第几个vector。调用next()的时候按照要求更新一下它们的值，另外记录一下总共有多少个元素以及当前已经取出了多少个元素，这样判断hasnext()只需要一个判断语句。

面试官问复杂度是多少，按照我的方法next()方法中判断要取第几个vector的元素复杂度是O(n)，这里的n是原先的vector的个数，因为某些vector可能已经取完了不能再取，所以要遍历一下找出下一个可以取的vector。他问能不能优化，我想用linkedlist(结果我不会用C++自身的linkedlist，只能自己写node\*然后搞出一个笨拙的链表)，每当有一个vector存满了就把linkedlist中指向这个vector的节点删掉，链表操作复杂度是O(1)的，所以整个next()函数的复杂度变成了O(1).

题目不难，但是细节很重要啊，唉，很多地方表现出自己基础不扎实.....

```
class ZigzagIterator {
public:
    ZigzagIterator(vector<vector<int>& list) {
        for(int i = 0; i < list.size(); ++i) {
            if(!list[i].empty()) q.push(list[i].begin(), list[i].end());
        }
    }

    int next() {
        auto t = q.front();
        q.pop();
        int res = *t.first;
        t.first++;
        if (t.first != t.second) {
            q.push(t);
        }
        return res;
    }

    bool hasNext() {
        return !q.empty();
    }
private:
    queue<pair<vector<int>::iterator, vector<int>::iterator>> q;
};

/** 
 * Your ZigzagIterator object will be instantiated and called as such:
```

```

* ZigzagIterator i(v1, v2);
* while (i.hasNext()) cout << i.next();
*/

```

<http://www.1point3acres.com/bbs/thread-306855-1-1.html>

11月14号来的电话面试。第一轮，给你一个2D array，求找出其中一个element的方法，然后替换当前element成一个新的element，以及给你一个start point和一个end point让你求能包含在内的所有元素之和，比如[[1, 2, 3], [4, 5, 6], [7, 8, 9]]，然后  $x_1 = 0, y_1 = 0, x_2 = 2, y_2 = 2$ ，就是返回  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$ ，follow up是如何能够快速求出他们的元素之和，我想了很久就是先构建出一个2D array然后里面存从起始点到ending point的所有元素之和，用的类似dp的思想，答得有点模棱两可，然后他说那是不是update element时间复杂度就高了呢，因为每次update都要修改你建的2D array，我说的是但是又找不出别的途径了。。感觉follow up答得有点不是很好= =

LC 308 Range Sum Query 2D - Mutable

第二轮，painting fence，先是给你N面墙，连续三面墙不可以一个颜色，如果只有两个颜色的话有多少种办法，我当时答了就是  $dp = dp[i-1] + dp[i-2]$ ，然后问我为什么，follow up 是有K个颜色怎么办，在他的提示下还是没能完全做出所以我是已经跪了，两周前面的之后也没来消息。。也没说有没有送到hc那里，给recruiter发邮件也没回

LC 276 Paint Fence//, LC 256 Paint House, LC 265 Paint House II (DP)

<http://www.1point3acres.com/bbs/thread-300222-1-1.html>

- Ph.D

第一轮：

1. 你有一个长方形，给定长和宽。输入是个字符串。要求找出最大的字号使得字符串可以塞进这个长方形里（可以换行）

你可以调用已有的两个函数。一个函数的参数是字符和字号，返回该字符在该字号下的宽度。另一个函数的参数只有字号，返回高度。也就是说每个字符在相同字号下的高度是一样的。另外还给你字号的范围（最大字号和最小字号，字号都是整数）。

2. 给你一个列表，每个元素都是指向某个结点的指针。每个结点都有左右两个相邻结点。所以有些结点是连着的。如果两个指针指向的结点之间的所有结点都被输入列表中的某个指针指向，那么我们就说两个指针在同一个连通区块中。求连通区块的个数。你只能通过输入数据中的指针间接访问那个链表的元素。

PS: 面试时第一轮第二题用的并查集的想法（由于只要求输出个数，其实代码很简单，不需要完整的并查集）。刚才又想了下，似乎还有个很简单的做法：先扫一遍，把所有被指向元素（的地址）加进一个哈希表。然后第二遍扫的时候利用哈希表判断每个元素是否是某个连通区块的最左边元素。类似lc上某个最长连续整数序列的题。

EX: 4个元素 0 <-> 1 <-> 2 <-> 3， <-> 表示他们左右连接。

然后输入是3个指针，第一个指针指向1，第二个指向3，第三个指向0.

那么0和1组成连接区块，3自己是一个连接区块，所以输出是2.

```

int connected(vector<ListNode*> ptr) {
    unordered_set<ListNode*> s;
    int cnt = 0;
    for(int i = 0; i < ptr.size(); ++i) {
        s.insert(ptr[i]);
    }
}

```

```

    for(auto it = s.begin(); it != s.end(); ++it) {
        if(s.count(it->prev) == 0) cnt++;
    }
}

```

## 第二轮：

1. 简单描述HashTable。
2. 假设有个程序一运行奔溃，你在代码里加了一行 打印，结果就莫名其妙不奔溃了。有什么可能的原因？
3. 貌似是经典的设计题： 用户一边输入，你要根据目前的输入推荐可能的补全。

<http://www.1point3acres.com/bbs/thread-299032-1-1.html>

## 第一轮。

1. 给定一个矩形，随机选择（均匀分布）矩形内的点，并返回该坐标。可以借助rand()函数（返回0~1随机数）思路大概是随机满足条件的选出x坐标，再随机选出满足条件的y坐标即可。

2. 接上。如果给定多个矩形呢？

思路：先选出矩形，再调用上面的函数。选矩形的时候可以借助rand()\*所有矩形面积，看随机出来的数落在哪个矩形上。另外有朋友提示，矩形可能重叠（我当时没想到）。这样的话，就把矩形分成小矩形就好了。

## 第二轮。

1. 给定一个3进制数的字符串，转成十进制数

```

int ternaryToDec(string s) {
    int res = 0, n = s.size();
    int idx = n - 1;
    int t = 1;
    while(idx >= 0) {
        int cnt = s[idx] - '0';
        res += cnt * t;
        t *= 3, --idx;
    }
    return res;
}

```

2. 给定一个字符串，返回含有k个不同字符的最长的子串。

## 加面

两个树判断是否相同，或者在swapping意义下相同。类似利口妖零零，不过交换左右子树之后相同也满足条件。感觉狗家确实每次面类型都不一样，如果有+面的话，着重准备之前没遇到过的类型吧

```

boolean isSimilar(TreeNode root1, TreeNode root2){

    if( root1 == null && root2 == null )
        return true;
    else if( root1 == null || root2 == null )
        return false;
}

```

```

    if( (root1.data == root2.data)
        && ( ( isSimilar(root1.left, root2.left) && isSimilar(root1.right, root2.right) )
            || ( isSimilar(root1.left, root2.right) && isSimilar(root1.right,
root2.left ) ) ) ){
        return true;
    }
    return false;
}

```

<http://www.1point3acres.com/bbs/thread-304782-1-1.html>

第一轮是个很nice的小哥，有点口音但是挺不出是哪里的。不是阿三。问了我简历的细节，然后笑着说，我只是随意问一下，不算入评分标准哒。

看到题目时候惊了，同义词那道题。最近好像很喜欢出这道题，之前随意翻面经的时候正好看到了。

Q1: 定义了一些同义词的pair，存在一个vector<pair>里，{{toyota, camry}，{good, great}}类似这种。然后给两句话判断是否同义。

如：“toyota is good”

“camry is great”是同义词。

同义词之间不能传递。

就unordered\_map<string, vector<string>>存储每个词的所有同义词。然后把两句话遍历，比较一下每个word，用了string的find函数找空格。

follow up: 就问了一下建map的复杂度啥的，可能比较菜，写完并没有时间问：如果同义词可以传递这种情况了。然后面试官表示答得挺好的，不知道是不是真心的QAQ。

第二轮本来是紧接着，因为面试官临时有会议，中间隔了一个小时。是个美国人，面试完全是鼓励式的，写一点，就说，good point。啥啥的。

Q2: 定义一个class来表示一个大整数，然后写一个两个整数相加的函数。默认这些数都是正的，并且是任意大小，会超出int/long的bit数。

用struct定义了一个vector<int>来表示前32bit，接下来32bit以此类推。

然后add函数，就是类似加法器。算sum和carry，进位。

最开始写了很多if比较两个数的vector size，后来优化了一下，如果超过size了就直接补0。

题是很简单，就是太菜了面试官让写test case然后才发现忘记加最高位carry进位了。

好吧那么简单的题我太对不起这题了。

<http://www.1point3acres.com/bbs/thread-302322-1-1.html>

第一题是double linkedlist，input是一木有排序 换装的双项链表，问能有几块连在一起的区域。弄懂题目后很简单，set可以秒。

第二题是给两个单词，问只swap一次后是不是一样。follow up 跟利特口 刘齐刘 有点像

<http://www.1point3acres.com/bbs/thread-307253-1-1.html>

第一轮是中国人老哥。题目是设计一个数据结构来存employee的关系。

比如A是B的direct manager，就有一种操作是<"set\_manager", "A", "B">。

比如B是C的同事<"set\_peer", "B", "C">

这时我们做这个操作<"query\_manager", "A", "B">，就会得到True，意思是A是B的manager（无论direct或者indirect）。

我们如果<"query\_manager", "A", "C">，这是也是True，因为C是B的同事，所以A也是C的manager。就是说同事关系有传递性。

有follow up，比如query的时候<"query\_manager", "A", "D">（D这个人没有初始化过），怎么办。有conflict怎么办，比如A是B的direct manager，E是C的direct manager，这是set\_peer(B,C)，就会有矛盾，B和C的direct manager应该是同一个人，E和A是两个人，这里有矛盾。

我说应该用union find或者dfs都可以实现，我先用比较蠢的方法写，每个employee都有一个set存他的同事，还有一个指针指向他的上司。

函数set\_manager(A,B)，把A设成B的上司，再把B的同事（set里面）的上司也设置成A。

set\_peer就互相加到set里面。

面试官提醒我说，好像我写这个没有peer的传递性。。但他跟我说时间不够了，你之前说过uf和dfs，就可以了。。意思是抬了我一手吧。。

Time O(N + MlogN) N employee, M operations

```
class Employee {
public:
    Employee(int N) {
        manager = vector<int>(N, -1);
        sz = vector<int>(N, 1);
        id = vector<int>(N, 0);
        for(int i = 0; i < N; ++i) id[i] = i;
    }
    void set_peer(int p, int q) {
        int pid = find(p);
        int qid = find(q);
        int mgr = max(manager[pid], manager[qid]);
        if(sz[pid] < sz[qid]) {
            id[pid] = qid, sz[qid] += sz[pid];
            manager[qid] = mgr;
        }
        else {
            id[qid] = pid, sz[pid] += sz[qid];
            manager[pid] = mgr;
        }
    }
    void set_manager(int p, int q) {
        int qid = find(q);
        manager[qid] = p;
    }
    bool query_manager(int p, int q) {
        while(1) {
            int qid = find(q);
            if(manager[qid] == p) return true;
        }
    }
};
```

```

        if(manager[qid] == -1) return false;
        q = manager[qid];
    }
    return false;
}

int find(int p) {
    while(p != id[p]) {
        id[p] = id[id[p]];
        p = id[p];
    }
    return p;
}
private:
vector<int> id;
vector<int> manager;
vector<int> sz;
};

```

第二轮美国人，问帕斯卡三角。

开始是求帕斯卡三角的任意一行一列的值，我用dp做了。求复杂度。

写这个def pascal(row, column): return一个int

然后followup是，如果要运行很多次pascal(row, column)，比如输出前N行帕斯卡三角，怎么优化？  
我做法是存一个全局变量，记录已经算好的值，就不用重新算了。

然后是，减少存储空间。

<http://www.1point3acres.com/bbs/thread-298973-1-1.html>

给一个单词字典，和一个起始字母，每次给字母长度加一并仍保证在字典中，求能产生的最长单词  
e.g., {"i", "im", "sim", "same"} and i -> "sim"

理解题目花了好久，开始以为新字母只能在后面加，说了个解法被大哥否定了。后来用map和queue做遍历，也不知道对不对

Time complexity O(N^2 \* k) level i to level i + 1 might have many edges.

```

bool oneAdd(const string &s1, const string &s2) {
    int i1 = 0, i2 = 0;
    bool flag = false;
    while(i1 < s1.size()) {
        if(s1[i1] == s2[i2]) {
            i1++;
            i2++;
        }
        else {
            if(flag == false) {

```

```

        i2++;
        flag = true;
    }
    else return false;
}
}
return true;
}

string findLongestWord(vector<string> words, string c) {
    unordered_map<int, vector<string>> dict;
    queue<string> q;
    unordered_set<string> visited;
    int max_len = 0;
    string res = c;

    q.push(c);
    for(auto word : words) {
        dict[word.size()].push_back(word);
    }

    while(!q.empty()) {
        string curr = q.front();
        q.pop();
        if(visited.find(curr) != visited.end()) continue;
        for(auto word : dict[curr.size() + 1]) {
            if(oneAdd(curr, word)) {
                q.push(word);
            }
        }
        if(curr.size() > max_len) {
            max_len = curr.size();
            res = curr;
        }
        visited.insert(curr);
    }

    return res;
}

```

<http://www.1point3acres.com/bbs/thread-304445-1-1.html>

第一个是白人小哥，上来直接做题 给一个graph和一个node 返回从这个node开始最多n个hop能到达的所有点的list  
followup是excat n个hop能到达点的list 中间要考虑各种情况，比如有cycle的时候

第二个听着像东欧大叔，上来扯淡20分钟因为google doc坏了 主要是聊我哦对research和之前的intern experience，然后口述做题，LC 140，不过是摩斯码，但本质完全一样，follow up如何提高效率，答memorization，又followup 如果一个摩斯码最长6位，怎么继续提高效率

不需要go over存摩斯码的hashmap，只要看长度为1，2，3，4，5，6的prefix是不是在hashmap里就行了

<http://www.1point3acres.com/bbs/thread-307266-1-1.html>

第一轮：

听上去是白人小哥。上来先问简历，挑了其中个项目问我。因为很久以前做的项目，说了好一会自己才慢慢回忆起来。bb了一会，小哥从项目中挑了几个问题来问我，比如：怎么判读两个单词之间的距离？之类的。大概10分钟吧，小哥说，咱来做题吧。

第一题，10进制转7进制，输入integer输出string。（我想应该算是很简单的题了吧）

```
int toSeven(int num) {
    int res = 0;
    int curr = 1;

    while(curr <= num) {
        curr *= 7;
    }
    curr /= 7;
    while(num > 0 || curr > 0) {
        for(int i = 6; i >= 0; i--) {
            if(num - curr * i >= 0) {
                num -= curr * i;
                res = res * 10 + i;
                curr /= 7;
                break;
            }
        }
    }
    return res;
}
```

第二题，给一个只含有数字【1到9】的矩阵，从矩阵中寻找九宫格(3x3的正方形，横着加，竖着加，对角线加都是15)，返回矩阵中这样的九宫格的个数。

上来一开始我有点懵，没啥思路，就说“哥你让我想想”。小哥说“没事，最basic的解法就行”，我说：“那我先bruce force做做，后面再优化行不？”，小哥说“没问题，整吧”

写的过程中犯了一个小错，小哥人超好，及时提醒了我一下。写完，小哥提示我，“你看每个九宫格有啥共同点？是不是中心都是15？这就可以优化了”，我一拍脑门“对啊！”

小哥：“没事儿，这题优化的空间也不大，我再问你几个小问题吧～”

于是follow up：你的这个程序里（我是Java写的）int[] 和 int[][] 有啥区别呀？内存里是啥样的呀？什么时候该用啥呀？都回答完还有点时间，小哥说“我们来bb吧”，bb了5分钟，完事。

=====

第二轮：

一个在狗家工作了9年大哥，听上去不像白人有点中东口音？人很nice。不过感觉他自己也一直在看note的感觉

上来先问Java基础知识（因为我之前选的用Java coding）

- abstract class 和 interface 的区别

- 能不能extend多个abstract class？

- abstract class能不能被instantiate。这里就是我自己煞笔了，说对不起我不确定，我觉得能，其实是不能。（自己基础知识不扎实）

完事儿做题。

第一题，给sorted array of integer，integers are range from 0 to 99. 输出所有0到99之间的空隙。比如：输入【1，2，3，50，51，89】输出“0，4-49，52-88，90-99”

### LC 163. Missing Ranges

这题其实也特简单，硬做。我是从第0个元素loop到最后做的。

follow up：如果元素特别多呢？我说：那就从0 loop 到99。大哥说“嗯，可以，下一题吧”

第二题，给两个string：str和pattern，判断是否match。比如：输入“abcdefg”和“acg”，输出true。因为acg在abcdefg里的相对位置都正确。如果是“abcdefg”和“adc”那就是不match。

### isSubsequence

//normal

```
bool isSubsequence(string s, string t) {
    if(s.size() == 0) return true;
    int ptrs = 0;
    for(int ptrt = 0 ; ptrt < t.size() ; ++ptrt){
        if(s[ptrs] == t[ptrt]){
            ++ptrs;
        }
    }
    if(ptrs == s.size())return true;
}
return false;
}
```

//If a lot query

```
bool isSubsequence(string s, string t) {
    if(s.size() == 0) return true;
    unordered_map<char, vector<int>> m;
    int n = t.size();

    //Preprocess
    for(int i = 0 ; i < n ; ++i){
        m[t[i]].push_back(i);
    }
    if(m.count(s[0]) == 0) return false;

    int prev = m[s[0]][0];
    cout << prev << endl;
    int i = 0;
    for(i = 1 ; i < s.size(); ++i){
        char c = s[i];
        if(m.count(s[i]) == 0) return false;
        if(m[c].size() == 0) return false;
        if(m[c][0] <= prev)
            prev = m[c][0];
        else
            return false;
    }
}
```

```

int l = -1;
int r = m[c].size();
while(l+1 < r){
    int mid = l + (r - 1) / 2;
    if(m[c][mid] > prev){
        r = mid;
    }else{
        l = mid;
    }
}

if(r == m[c].size())return false;
prev = m[c][r];
}
if(i == s.size()){
    return true;
}
else{
    return false;
}
}
}

```

运气真的很好，感觉这题也很简单，不知道有没有坑。两个指针做的。

做完和大哥bb了10分钟，结束

<http://www.1point3acres.com/bbs/thread-307200-1-1.html>

应该是名字清奇的research-oriented Ph.D.intern岗，看地里少有专门针对这个职位的面经，贡献一下，多吐槽hhresearch intern两轮背靠背电面，第一轮是正常的技术面，第二轮是research面，会尽量根据你的expertise找对应面试官。

第一轮：一个曾经也做过research的SDE，native speaker，号码是山景城的。他说是现在还在google做一些和UI/UX research有关的工作。

我明明填的python，他一上来就问我能不能用js面，我当时非常懵逼，因为我没在白板上写过js，各种api每次忘了都是Stack Overflow.....后来得知是因为他自己用js和c++并且知道我老板（顺带着知道我们组会用js做不少东西）。

没有自我介绍，直接做题：

输入：一个高为h，宽为w的屏幕，一行字符串str，字号范围(MIN, MAX)，辅助api: int widthOf(char c, int fontSize), int heightOf(int fontSize)←同字号的字等高不一定等宽

输出：最大的可以把str fit进屏幕的字号

注意：不需要做复杂的line breaking，比如“hello world”，不需要必须在空格换行，任意位置都可以

我很简单的用了二分找字号，然后在算每一个字号能不能fit的时候，丑陋的选择了for loop每个char，如果可以放在当前行就放，放不下就停在这个char之前换行，直到用完char或者到达了最大行数(w/heightof)。

到这里这个面试还很简单，然后就开始直奔玄学，小哥觉得我的for loop太丑陋，因为在iterate字符的时候一直是i++，但是如果一旦放不下了需要i--一下保证当前char会被保留，这样就在for loop改了i，问我有没有别的办法。我说了几种implement的trick明显都不是他要的，最后他无奈了，提示说有的时候不同char的width是一样的，我就说一样width就可

以直接批量除了不是吗，他说对的，然后就没有问下去了，问我有没有问题想问。我有点崩溃的问小哥能不能告诉我理想算法应该是啥，他说不行2333

```
int widthOf(char c, int fontSize){}
int heightOf(int fontSize){}

bool canFit(string s, int width, int height, int fontSize) {
    int curW = 0, rowNum = height / heightOf(fontSize), idx = 0;
    if(rowNum <= 0) return false;
    while(idx < s.size()) {
        int charW = widthOf(s[idx], fontSize);
        if(charW > width) return false; // even one char can't fit
        if(curW + charW <= width) {
            curW += charW;
        } else {
            curW = charW;
            rowNum--;
            if(rowNum <= 0) return false;
        }
        idx++;
    }
    return true;
}

int find(vector<string>& strs, int width, int height) {
    if(strs.empty()) return true;
    int l = minFontSize, r = maxFontSize;
    string sentence = "";
    for(int i = 0; i < strs.size(); ++i) {
        sentence += strs[i];
        if(i != strs.size() - 1) sentence += " ";
    }
    while(l < r) {
        int mid = l + (r - l) / 2;
        if(canFit(sentence, width, height, mid)) l = mid;
        else r = mid - 1;
    }
    return l;
}
```

我就改问他也做过research也去过industry，觉得research的经历对他现在的工作最大的帮助是啥，觉得区别是啥，然后热情的小哥给我说了好多，一直说到第二个电话打进来被迫挂掉.....

划重点，特别感谢这位小哥，感觉他肯定是抬了一手，毕竟我几乎没刷过题，没用js写过白板，还是人生中第一个tech面试，感觉答成啥样心里有数==

第二轮：一个BOS google brain的researcher，听起来是个native大叔。

1. 照着resume让我讲了三个我的项目，穿插着和项目有关的问题，比较像1v1版的conf presentation。

2. 让我回答觉得自己为什么适合到google brain，有没有什么很想做的project，自己的expertise为何和google可以衔接。我就高度赞美了狗家的research，又硬扯了一下他没问到的一个项目说相关，他竟然也觉得相关，然后我就很努力的把这个其实刚起头的项目扯成我已经做了一段并且知道自己在做啥的样子。

3. 问他问题，我就问他们组到底在干啥（最开始自我介绍他简短的说了一下），说完以后我rephrase了一下他的话表示理解和赞扬，然后这个小时就结束了。

<http://www.1point3acres.com/bbs/thread-306874-1-1.html>

1轮是个白人小哥，大概意思就是把26个字母按顺序按给定宽度组成一个矩阵，比如输入宽度为5那就是  
A B C D E

F G H I J

K L M...

然后给一个String，比如BIG，那么输出 R & RRD & LL &，就是再矩阵上怎么move

然后这题秒了，问了一些简单的followup，然后又出了一个题判断一个int是不是2的幂，秒了  
然后小哥说他没题了，然后就跟他扯了20分钟淡。

2轮好像是个ABC小姐姐，迟到了15分钟。。上来先问了一堆java基础，比如int最大值是多少，给一个int赋一个超大的值会怎么样

然后做题，就是设计一个class 叫BigInteger, 能对超大的int做 + - \* /，

楼主本来EE背景，一直跟加法器乘法器打交道，但是看到除法就懵逼了，除法要咋弄。。。但是后来小姐姐说我们先实现一下加法。。。

然后就开始设计class和实现加法，期间一直和她聊这个class的细节，很费时间，然后好多地方得跟她解释我为啥这么做。最后写完时间就差不多了，然后就结束了。

<http://www.1point3acres.com/bbs/thread-305839-1-1.html>

一轮 没做过，猜词游戏 给一个词，给一个猜的词然后返回有多少字母对了，有多少字母位置不对但是也在给的词里。

这轮白人小哥放水，过一遍词直接hash解决，没有给followup（本来要给的后来决定不问了。。。）

二轮 老人一脸黑，上来LC 31 Next Permutation 然后问了如何设计一个tape 然后要用stack来存数据

就是你写一个class来represent一个tape，所以就是有个head然后data存在一个长条上，head的位置可以左右移，每个位置存着一个data，然后让你支持的operation有往左移，往右移，读取，写入。其实就是disk，只不过不是圆形的而是长条的。还有可存储数据没有数量限制

Two stack

S1 : store data, S2 buffer

```
moveRight(int x) {
    if(head.size() > x) {
        while(x--) {
            back.push(head.top());
            head.pop();
        }
    } else {
        moveLeft(N - x);
    }
}
```

```

}

moveLeft(int x){
    if(back.size() >= x) {
        while(x--) {
            head.push(back.top());
            back.pop();
        }
    } else {
        moveRight(N - x);
    }
}

Data read() {
    return head.top();
}

bool write(Data d) {
    head.push(d);
    N++;
}

```

<http://www.1point3acres.com/bbs/thread-304617-1-1.html>

最近店面加试，两道题，第一题类似有给两个数组，返回所有两个数组之间元素交叉相乘的总和  
第二题是判断一个人图是否为二分图

```

bool solve(int V, vector<vector<int>>& G) {
    vector<int> color(V, 0);
    for(int i = 0; i < V; ++i) {
        if(color[i] == 0){
            if(!dfs(i, 1, G, color));
                return false;
        }
    }
    return true;
}

bool dfs(int v, int c, vector<vector<int>>& G, vector<int>& color) {
    color[v] = c;
    for(int i = 0; i < G[v].size(); ++i) {
        int u = G[v][i];
        if(color[u] == c) return false;
        if(color[u] == 0 && !dfs(u, -c, G, color)) return false;
    }
    return true;
}

```

<http://www.1point3acres.com/bbs/thread-299194-1-1.html>

两个面试back to back。第一个是个国人小哥。一开始我们的Google doc不同步，管recruiter重新要了一个，浪费了好几分钟。所以不聊天直接开始做题，幸好题目很简单：

给一个integer list in ascending order，把每个数都平方，返回排好序的平方数组。例子：[-2, -1, 0, 2]，返回[0, 1, 4, 4]。

做完还有时间就又聊了聊简历上的项目。

### LC 360. Sort Transformed Array

第一场面试还有个follow up忘记说了，不是把每个数都平方，而是 $y = x^2 + a*x$ ，再返回按这个y值排好序的数组。

第二个人出了两题，

第一题给两个string，判断能否通过交换string1中的两个字符得到string2。

第二题给一堆synonyms，[("restaurant", "cafe"), ("ratings", "reviews"), ...]，再给一些queries[("restaurant ratings ...", "cafe reviews"), ...]，要求返回每个query里的对应词是否都是synonym。两题做完时间刚好用完。

<http://www.1point3acres.com/bbs/thread-305663-1-1.html>

1. 输入一个由黑白格子组成的正方形grid/matrix, 求黑色格子连成的线段的最大长度（包括vertical, horizontal, diagonal）。面试官貌似是个美国小哥，话不多，没啥提示。

### LC 562. Longest Line of Consecutive One in Matrix

2. 给定一些变量之间的关系，比如 $0.5*A=B$ ,  $0.6*B=C$ , ...,  $0.2*D=E$ , 然后输入任意两个变量名，输出这两个变量的关系，比如 $*E=A$ ，基本上就是个graph search。面试官是个中国人，人超好，回答问题很耐心。

### LC 399

<http://www.1point3acres.com/bbs/thread-305928-1-1.html>

第一场：面试官一上来问了好多BQ，感觉他十分看重communication和collaboration，看重你如何表述自己的idea，并且让你的team member明白。

算法题就是如何判断一棵树是不是平衡二叉树，接下来问如何判断是不是perfect二叉树，是不是complete二叉树。感觉自己有点紧张很多想法也没有表述很清楚，还是有些遗憾吧~

Balanced: difference of left and right subtree is less than 1.

Perfect: full (every node except leaf has two children) and complete (every level except last level is full, all leave are as left as possible)

```
int findADepth(Node *node)
{
    int d = 0;
    while (node != NULL)
    {
        d++;
        node = node->left;
    }
    return d;
}

/* This function tests if a binary tree is perfect
or not. It basically checks for two things :
1) All leaves are at same level
2) All internal nodes have two children */
bool isPerfectRec(struct Node* root, int d, int level = 0)
{
    // An empty tree is perfect
    if (root == NULL)
```

```

    return true;

// If leaf node, then its depth must be same as
// depth of all other leaves.
if (root->left == NULL && root->right == NULL)
    return (d == level+1);

// If internal node and one child is empty
if (root->left == NULL || root->right == NULL)
    return false;

// Left and right subtrees must be perfect.
return isPerfectRec(root->left, d, level+1) &&
       isPerfectRec(root->right, d, level+1);
}

// Wrapper over isPerfectRec()
bool isPerfect(Node *root)
{
    int d = findADepth(root);
    return isPerfectRec(root, d);
}

```

第二场：面试官问了问简历就开始做题啦，面试官引导着理解题目给例子，好久以前的面经题，也没刷过，只能硬刚。输入一个符合文法的字符串，输出计算数值，所有输入之间以空格分割。

`expr ::= int | '(' op expr+ ')'`  
`op ::= '+' | '*' | '-' | '/'`

举了几个例子：

"3" -> 3  
"(+ 1 2)" -> 3  
"(+ 3 4 5)" -> 12  
"(+ 7 (\* 8 12) (\* 2 (+ 9 4) 7) 3)" -> ...

用了stack来实现，google doc写代码体验好差差差，follow up是如果有个占用时间很多的操作符\$，如何优化？反正我也没搞明白也木有答上来。

<http://www.1point3acres.com/bbs/thread-305005-1-1.html>

面经：

1.1. group anagrams, 见 LC

1.2. 有点像hit counter那道题，见LC，按时间顺序有 `put(key, value, duration)`, `get(key)`，key 会过期，过期就没有这个对应的record了，然后return null。我问她要了个getCurrentTime API

2.1 findKthElement in a BST

In-order traversal, O(N) worst case

2.2 followup, BST node里多加一个属性是#ofchildren，findKthElement

Time complexity is O(height of BST)

2.3 给 a list of sentences, 然后给一个单词 , predict 它后面的一个单词 , 就随便讨论了讨论 , 我说hashmap, 他说行 , 就没让写

<http://www.1point3acres.com/bbs/thread-305381-1-1.html>

一面感觉是个三哥 , 给一棵树 , 每个节点有一个parent节点 , 要求删掉一个子树

二面感觉是个白人小哥 , next greater element.

<http://www.1point3acres.com/bbs/thread-218469-1-1.html>

interview 1 : 给一个array, 是以max-heap形式的。比如100 , 40 , 20 , 2,30 , 15 , 9 等等... 返回第k大的数。

```
100  
5      50  
3  2    20  30
```

Pq里面一开始放的是root , 每一次pop操作把pop出去的数的children加到pq中

O(klogk)

interview 2: 被鸽子了.....555555 , 是安排下下周好还是今天好呢...都等困了

补充内容 (2017-1-7 04:55):

哦对了 , 题意还有不能改变array的顺序

补充内容 (2017-1-27 07:07):

第二道题 :

You are given an array of numbers and a positive integer k. You are allowed to move an element of the array at most k position earlier, or any number of positions later. Given this const...

补充内容 (2017-1-27 07:07):

Given this constraint, you need to make the array as sorted as possible.

For example, if k=2 and the input array is 5 4 2 3 1, then the output should be 2 3 1 4 5.

```
vector<int> asSorted (vector<int>& nums, int k) {  
    priority_queue<int> pq;  
    vector<int> res;  
    for(int i: nums) {  
        pq.push(i);  
        if(pq.size() > k) {  
            res.push_back(pq.top());  
            pq.pop();  
        }  
    }  
    while(!pq.empty()) {  
        res.push_back(pq.top());  
        pq.pop();  
    }  
    return res;  
}
```

<http://www.1point3acres.com/bbs/thread-302897-1-1.html>

第一轮，给两个字符串，说这两个字符串的唯一区别是一个比另一个多了一个字母，问多了哪个字母。比如"abc"和"abac"。这个题可以说是很简单了，用个数组计出现次数就好了。**XOR**

第二轮，应该是跪在这上边了。

给一堆区间，比如 [-1.1, 1.0], [-0.5, 3.5], [3.6, 4.0], ...，再给一个点target，比如0.1，要返回所有包含了这个点的区间。

最简单的做法就是一个一个区间比较，返回所有包含了target的区间，这是linear time。

又问，可以对这些区间做pre-processing，pre-processing只做一次所以复杂度不计。问怎么做可以比linear time更快。

和面试官讨论的大致思路是，按照区间左端点做sorting，然后binary search。但是因为区间有overlap种种问题，最后也没做出来。

<http://www.1point3acres.com/bbs/thread-225024-1-1.html>

第一轮面试应该是个美国人，在google工作已近十多年了。题目大致是给定一个二叉树，有其中一条枝连错了，让我找到这个错误的枝的node，删掉这个node并返回根节点。

Every node has at most one parent

LC685 下面的解答是错的

```
TreeNode* deleteWrongNode(TreeNode* root) {
    unordered_set<TreeNode*> visited;
    return findInvalidNode(root, visited);
}

void findInvalidNode(TreeNode* root, unordered_set<TreeNode*>& visited) {
    if(!root) return;
    visited.insert(root);
    if(root->left){
        if(visited.count(root->left) == 0){
            root->left = findInvalidNode(root->left, visited);
        }
    } else {
        return nullptr;
    }
}
if(root-> right) {
    if(visited.count(root->right) == 0) {
        root->right = findInvalidNode(root->right, visited);
    }
} else {
    return nullptr;
}
}
return root;
}
```

第二轮的题目是一个二维数组，每个点是1或者0。找到所有的1使得这个1是它所在的行和列的唯一的1。

LC 531 Lonely pixel

<http://www.1point3acres.com/bbs/thread-304467-1-1.html>

## 第一题

给出一个同义词的列表，如{great, fine}, {acting, drama}, {skills, talent}

问两个句子是否同义，如"great acting skills", "fine drama talent"是同义的，"great great"和"fine great"也是同义的  
句子同义的定义是：两个句子里面的单词按次序两两同义。没有传递性（比如great和fine同义，fine和good同义，但great和good不一定同义

直接做就可以了，用个unordered\_map<string, unordered\_set<string>>存下每个单词的同义词列表。

Followup: 有传递性

做个dfs就可以了。O(n^2)

## 第二题

给出公司并购的关系列表，比如

```
[  
["baidu", "ofo"],  
["mobike", "alibaba"],  
]
```

表示baidu并购了ofo，摩拜并购了阿里巴巴。。。

求最长的一个并购链。保证无环。

就是求有向无环图的最长路径，直接O(n)的记忆化DP就可以。

这题的followup都很简单。。不提。

Longest path in DAG: topological sort -> dp

求长度，topological sort(这段是bfs写的，慢了)

```
int longestPath(vector<pair<string, string>> pairs) {  
    unordered_map<string, vector<string>> graph;  
    unordered_map<string, int> inDegree;  
    queue<pair<string, int>> q;  
    int res = 0;  
  
    for(auto p : pairs) {  
        graph[p.first].push_back(p.second);  
        inDegree[p.second] += 1;  
        inDegree[p.first] += 0;  
    }  
  
    for(auto node : inDegree) {  
        if(node.second == 0) {  
            q.push(make_pair(node.first, 1));  
        }  
    }  
  
    while(!q.empty()) {  
        string node = q.front().first;  
        int dis = q.front().second;
```

```

q.pop();

res = max(res, dis);

for(auto neighbor : graph[node]) {
    q.push(make_pair(neighbor, dis + 1));
}
}

return res;
}

求path, dfs
class Solution {
public:
    int max_dis;
    vector<string> max_path;
    void helper(unordered_map<string, vector<string>> &graph, vector<string> &path, string start, int dis) {
        if(graph[start].size() == 0) {
            if(dis > max_dis) {
                max_path = path;
            }
            return;
        }
        for(auto neighbor : graph[start]) {
            path.push_back(neighbor);
            helper(graph, path, neighbor, dis + 1);
            path.pop_back();
        }
    }
}

vector<string> longestPath(vector<pair<string, string>> pairs) {
    unordered_map<string, vector<string>> graph;
    unordered_map<string, int> inDegree;
    vector<string> starts;
    vector<string> path;
    max_dis = 0;

    for(auto p : pairs) {
        graph[p.first].push_back(p.second);
        inDegree[p.second] += 1;
        inDegree[p.first] += 0;
    }

    for(auto node : inDegree) {
        if(node.second == 0) {
            starts.push_back(node.first);
        }
    }
}

```

```

    for(auto start : starts) {
        path = vector<string>({start});
        helper(graph, path, start, 1);
    }

    return max_path;
}
};


```

个人体验：

哎。。。面跪了。第一题是个印度姐姐，人很nice可是我有些地方听不太懂。最后followup还一时脑残用了O( $n^3$ )的floyd算法（因为当时懒得写dfs.....），最后被问到才发现我为什么不写平方的dfs啊！！！跪了跪了

<http://www.1point3acres.com/bbs/thread-297584-1-1.html>

Interview1:

上来大概聊了一下项目，这轮题目还比较简单，基本就是连通问题，就是矩阵判断是否连通, 1 path, 0 wall。之后的followup是返回连通路径。自己作死，写第二个的时候很自信，结果出了bug。

小哥没什么口音，人超好。

Interview2:

直接开始做题，给出一个与图有关的game，n个房间，每个房间可以通往另外随机个房间，但是有的门会有锁，需要对应的钥匙，一个房间可以是空的或者有一把钥匙，找出终点。

最开始让自己设计数据结构，我用的邻接链表，之前没做过类似的题目，和小哥在细节实现上探讨浪费了不少时间，这其实人家是不关心的。之后，让写一个方法验证生成的图是否是有解的，就用bfs和hashmap实现了一下。

<http://www.1point3acres.com/bbs/thread-302806-1-1.html>

Google暑期实习面试，两轮背靠背

第一轮是个小哥，上来问了一个比较简单的问题。

1 ) 输入是两个string s1 和 s2，之间就差一个extra character，要求输出这个extra char，地里原题，利用XOR就能做了。

2 ) follow up是shuffle其中一个string，但是已经是用XOR了，所以这个题目就直接pass了

3 ) 再follow up就是说数据量太大怎么办，答MapReduce，又问bottleneck在哪儿，思考了一下说是如果只有一个reducer的话那个reducer就是bottleneck，解决方法就是multi-layer mapreduce

第二轮是个大姐，上来让我介绍了一下自己的research，之后就上题目

1 ) 输入是一个List<Message>，每个Message由<usr, text>组成，问top n word count usr，也就是输出List<String>，list里面是n个usr。我的解法是hashmap统计词频，再用minheap排序，问了一下时间复杂度

2 ) followup，在提示下进行了一下优化，每次插入minheap的时候，如果minheap size到了N，先判断一下队列头的元素与当前元素的word count的对比，如果是大于关系则根本不需要插入到minheap中

```

struct Compare {
    bool operator()(pair<string, int>& a, pair<string, int>& b) {
        return a.second > b.second;
    }
};

```

```

vector<string> topNwordCount (vector<Message>& list) {
    unordered_map<string, int> m;
    for(Message msg: list) {
        m[msg.usr] += msg.text.size();
    }
    priority_queue<pair<string, int>, vector<pair<string, int>>, Compare> pq;
    for(auto p: m) {
        if(!pq.empty() && pq.size() == N && pq.top().second < p.second) pq.pop();
        pq.push(make_pair(p.first, p.second));
    }
    vector<string> res;
    while(!pq.empty()) {
        res.push_back(pq.top().first);
        pq.pop();
    }
    reverse(res.begin(), res.end());
    return res;
}

```

<http://www.1point3acres.com/bbs/thread-305693-1-1.html>

第一轮，LC 124 相似，但更简单，只用考虑从root开始的path

第二轮，LC 524 原题。。wording不太一样，找dictionary里最长的word，是string的subsequence。

<http://www.1point3acres.com/bbs/thread-303077-1-1.html>

第一轮：

利口药企三，我用的stack

follow up：优化空间，用2个 point cur ,prev.(Morris Inorder Traversal)

```

class BSTIterator {
public:
    TreeNode* curr;
    BSTIterator(TreeNode *root) {
        curr = root;
    }

    /** @return whether we have a next smallest number */
    bool hasNext() {
        if(curr != nullptr) return true;
        return false;
    }

    /** @return the next smallest number */
    int next() {
        while(curr != nullptr) {
            if(curr->left == nullptr) {

```

```

        int res = curr->val;
        curr = curr->right;
        return res;
    }
    TreeNode *prev = curr->left;
    while(prev->right != nullptr && prev->right != curr) {
        prev = prev->right;
    }
    if(prev->right == nullptr) {
        prev->right = curr;
        curr = curr->left;
    }
    else {
        int res;
        prev->right = nullptr;
        res = curr->val;
        curr = curr->right;
        return res;
    }
}
return -1;
}
};


```

第二轮：

1. You are given the following helper function:

doubleRandBetween(double min, double max);

randBetween(1.3, 5.0) = 4.3

randBetween(1.3, 5.0) = 3.1

给一个矩阵以及randBetween API，返回矩阵中任意一个点。

class Rectangle {

public:

Rectangle(int x1, int y1, int x2, int y2): x1(x1), y1(y1), x2(x2), y2(y2) {}

double getArea() {

return sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2));

}

pair<double, double> getRandomPoint() {

pair<double, double> res = {randBetween(x1, x2), randBetween(y1, y2)};

return res;

}

private:

int x1, y1, x2, y2;

};

Rectangle is (x1, y1) left bottom (x2, y2) right top

x = doubleRandBetween(x1, x2);

```

y = doubleRandBetween(y1, y2);
2. You're given N rectangles. Generate a point that is uniformly random from the set of given rectangles. 矩阵大小不一样
, 选中矩阵的概率也不一样。
randomPoint([recA, rec B, rec C]) = {1, 1} ← point in rectangle C
randomPoint([recA, rec B, rec C]) = {1, 4} ← point in rectangle A
randomPoint([recA, rec B, rec C]) = {2, 4} ← point in rectangle A
randomPoint([recA, rec B, rec C]) = {3, 3} ← point in rectangle B

pair<double, double> randomPont(vector<Rectangle>& list) {
    double total = 0.0;
    int n = list.size();
    vector<double> randomRec(n, 0.0);
    for(int i = 0; i < n; ++i) {
        total += list[i].getArea();
        randomRec[i] = total;
    }
    double target = randBetween(0, total);
    auto it = upper_bound(randomRec.begin(), randomRec.end(), target);
    int rec = it - random.begin();
    return randomRec[rec].getRandomPoint();
}

```

<http://www.1point3acres.com/bbs/thread-298713-1-1.html>

第一轮，写一个迷宫的类，包括接口和成员变量，迷宫里有起点和重点。

然后写一个函数从起点走到重点，返回经过的点。数据结构都自己写。

第二轮，阿三哥，人很好，

两题，第一题，虽然是昨天刚做的，但是记不清了=。=总之是warm up，不用担心。

第二题是 **LC 340. Longest Substring with At Most K Distinct Characters**, 小哥来来回回提醒还是没写完，其实很简单，有些紧张，一声叹息。move on了。

<http://www.1point3acres.com/bbs/thread-298693-1-1.html>

头两轮店面已经是10.6的事儿了：

10.6 -----

第一轮：**LC 208 Implement Trie (Prefix Tree)**，搜索框预测，之前面经看到，面试前还复习了下，感谢地里的小伙伴，国人小哥也很nice，感觉不错，follow up是如何预测形近词（dog->dig），LZ说加个近似度，小哥说挺好的。

第二轮：目前还没找到原题，应该是美国哥，先问简历，然后问迄今最挑战的是什么（现在感觉是这次的电面lol），然后说了自己的research，提到SVM，小哥问我是什么。。。问题是比特array，用python实现，LZ表示一个没有CS背景的我完全没有听过，沟通是到底是啥就花了20多分钟，加上闲聊的10来分钟（大家自我介绍从简为好），最后只实现了一个function，然后弱弱的能不能加个题挽回下，小哥说去找你的recruiter。

我真的发邮件给recruiter说，很感谢然后第二轮面的不好能不能再给轮Bonus Interview，不出所料，当时并没有回。LZ最近比较忙，只好去老老实实改论文了。然而，到了10.12，竟然发邮件说HC需要one more piece of data，然后约了今天的见面，Thank God！

10.17 -----

加面 第三轮：是个三哥，人真的很好，上来说下流程就开始介绍题（面完看@cser017 的昂赛面经才知道是 LC 399，是各种进制转换，哎，面之前只刷了最近的电面经，没想到。。。大家刷的话真得都看），<--怎么叨叨了这么多>\_<"

LZ先说用BFS，三哥说不好，换一种，想到了dict但是还是思路不对，三哥第一次提示选个中间元素，LZ第一次豁然开朗，用了两个字典，然后实现功能，写的时候就发现不对，三哥第二次提示加个字典存对应元素，LZ第二次豁然开朗lol。写完，三哥给了个 testcase，没等提示，LZ第三次拍大腿，把已有的结构改了一下，然后三哥第一次说了句顾得。以为没事了，结果问时间复杂度的时候，LZ抽筋的说了O(n)，三哥没再提示（不知道会不会影响太多）。然后问问题，互相道谢，LZ问了一遍三哥的名字，表示灰常感谢。

Time: O(V+E)

Space: O(VE)

```
class Solution {
public:
    vector<double> calcEquation(vector<pair<string, string>> equations, vector<double>& values, vector<pair<string, string>> queries) {
        int n = equations.size();
        vector<double> res;
        if (n == 0) return res;
        unordered_map<string, vector<string>> edge;
        unordered_map<string, vector<double>> weight;
        unordered_set<string> visited;

        for (int i = 0; i < n; ++i) {
            string u = equations[i].first;
            string v = equations[i].second;
            double w = values[i];
            edge[u].push_back(v);
            weight[u].push_back(w);
            edge[v].push_back(u);
            weight[v].push_back(1/w);
        }

        for (int i = 0; i < queries.size(); ++i) {
            string start = queries[i].first;
            string end = queries[i].second;
            if (edge.find(start) == edge.end() || edge.find(end) == edge.end()) {
                res.push_back(-1.0);
            } else if (start == end) {
                res.push_back(1.0);
            }
            else {
                double t = dfs(start, end, 1.0, edge, weight, visited);
                res.push_back(t == 0.0 ? -1.0 : t);
            }
        }
    }
}
```

```

        }
    }
    return res;
}
double dfs (string u, string end, double cur, unordered_map<string, vector<string>>& edge, unordered_map<string,
vector<double>>& weight, unordered_set<string>& visited) {
    if (visited.find(u) != visited.end()) return 0.0;
    if (u == end) {
        return cur;
    } else {
        vector<string> neighbor = edge[u];
        vector<double> value = weight[u];
        int n = neighbor.size();
        visited.insert(u);
        double t = 0.0;
        for (int i = 0; i < n; ++i) {
            t = dfs(neighbor[i], end, cur * value[i], edge, weight, visited);
            if (t != 0.0) {
                break;
            }
        }
        visited.erase(u);
        return t;
    }
}
};


```

<http://www.1point3acres.com/bbs/thread-303518-1-1.html>

1.given a binary tree, delete the bad edge. follow up given a binary search tree

bad edge要想，哪些情况。一面估计跪了。

LC Redundant Connection

2. given a target node in a directed graph, find the shortest cycle including this node, return the whole path.

输入可以自己随便设计。

国人姐姐全程不交流不给提示。。愣是最后做出来了。。但是不是最优解，空间复杂度高了点。但是也要怪我，才知道面试hint要自己去要的。

```

struct Node
{
    string label_;
    unordered_set<Node*> adjacents_;
    Node(const string& label) : label_(label) {}
    void link(Node* v) { adjacents_.insert(v); }
};

vector<Node*> shortest_cycle_path(Node* s)
{

```

```

unordered_map<Node*, Node*> parents;
deque<Node*> q{ s };

while (!q.empty()) {
    Node* u = q.back();
    q.pop_back();
    for (auto v : u->adjacents_) {
        if (parents.count(v) > 0) continue;
        parents[v] = u;
        if (v == t) {
            q.clear();
            break;
        } else {
            q.push_front(v);
        }
    }
}

// recover path
vector<Node*> path;
auto current = s;
do {
    path.push_back(current);
    current = parents[current];
} while (current != nullptr && current != t);
if (current != nullptr) path.push_back(t);
if (path.size() == 1) path.clear(); // no cycle
reverse(path.begin(), path.end());
return path;
}

```

<http://www.1point3acres.com/bbs/thread-304235-1-1.html>

第一轮：印度大哥，国际惯例迟到五分钟，给你两个vector a, b，告诉你b是a的permutation，让你返回这个permutation，具体返回啥自己定义，反正你返回的这个东西，还得能用到别的vector上，比如你返回个P，如果我再给你个c，你能用这个P把c变成相应的permutation d，还让写了test。感觉大哥题目表述的不是很清楚，问他是不是想要这个，他就支支吾吾的反正感觉他自己都不知道想要问什么。。。

第二轮：国人小哥，很给力，第一题找 x 的square root，x 是float number。第二题给你一堆2D的points，找出他们所能形成的rectangle（用四个点）的最大面积。

```

float squareRoot(float n)
{
    /*We are using n itself as initial approximation
     *This can definitely be improved */
    float x = n;
    float y = 1;
    float e = 0.000001; /* e decides the accuracy level*/
    while(x - y > e)

```

```

{
    x = (x + y)/2;
    y = n/x;
}
return x;
}

struct Point {
    int x;
    int y;
    Point(int x, int y) {
        this->x = x;
        this->y = y;
    }

    bool operator==(const Point &p) const {
        return this->x == p.x && this->y == p.y;
    }
};

struct hasher {
    size_t operator()(const Point &p) const {
        return hash<int>()(p.x) ^ (hash<int>()(p.y) >> 1);
    }
};

int maxRectangle(vector<Point> points) {
    unordered_set<Point, hasher> dict;
    int res = 0;

    for(auto p : points) {
        dict.insert(p);
    }

    for(int i = 0; i < points.size(); i++) {
        for(int j = i + 1; j < points.size(); j++) {
            if(points[i].x != points[j].x && points[i].y != points[j].y) {
                Point p1(points[i].x, points[j].y);
                Point p2(points[i].y, points[j].x);
                if(dict.find(p1) != dict.end() && dict.find(p2) != dict.end()) {
                    res = max(res, abs(p1.x - p2.x) * abs(p1.y - p2.y));
                }
            }
        }
    }
}

return res;
}

```

第三轮（等了一个礼拜后通知加面）：国人大哥，一上来就跟我讲中文，搞得我措手不及。先摔了一个 add two string，就是给你两个string，代表两个数，求他们和，秒了之后大哥问我这题是不是做过，我说做过类似的吧，然后大哥又给了我一道题：有个n个村庄，在一条线上，现在要在这些村庄里建K个医院，求这些村庄到这K个医院的最小距离。这个题我似曾相识，但是死活想不起来怎么做，求提示后大哥说用dynamical programming，反正我最后也没写出来这个状态转移方程，跪

POJ 1160

$dp[i][j]$  : minimum distance build j hospitals in i villages

$dist[i][j]$  : minimum distance build 1 hospital between village i and j, always pick mid =  $(i + j + 1) / 2$

$$dp[i][j] = \min(dp[k][j-1] + dist[k+1][i]) \quad 0 \leq k < i$$

```

int village[MAX_N]
int dist[MAX_N][MAX_N]
int dp[MAX_N][MAX_K]

int N, K, S

void preprocess() {
    sort(village.begin(), village.end());
    for(int i = 1; i <= N; ++i) {
        for(int j = i; j <= N; ++j) {
            int mid = (i + j + 1) >> 2;
            int sum = 0;
            for(int k = i; k <= j ; ++k) {
                sum += abs(village[k] - village[mid]);
            }
            dist[i][j] = sum;
        }
    }
    dp[0][0] = 0;
    memset(dp, 0x3F, sizeof(dp));
}

int compute() {
    for(int i = 1; i <= N; ++i) {
        for(int j = 1; j <= K; ++j) {
            for(int k = 0; k < i; ++k) {
                dp[i][j] = min(dp[i][j], dp[k][j-1] + dist[k+1][i]);
            }
        }
    }
    return dp[N][K];
}

```

<http://www.1point3acres.com/bbs/thread-305201-1-1.html>

第一轮 一串排好序的只有大写字母的字符串找出其中字母B的个数 二分求解

第二轮 很奇怪的一道题，一共两部分，要找一个string重复子串的字母个数

比如ABCABCABCABC, 重复子串可以是ABCABC (\* 2)也可以是ABC (\* 4)

第一部分是写一个Helper function判断数字i是不是一个符合条件的重复子串个数，比如上面那个例子6, 3都应该是true .

第二部分是写一个function找最短的重复子串的字符数，比如上面的例子返回3

第二部分除了brute force并不知道怎么做，第一反应是二分结果写了一半发现不行... 讨论了一会儿用哈希表过一遍string

找到每个字母的个数，找到最小的那个假设为k，所以任何子串做多重复了k次，最短重复子串就是字符串总长度除以k...

然后一直用第一问写的helper。。。感觉很奇怪肯定不是标答，不过面试官说还行...写完跟面试官聊了很久感觉

interaction其实很好

<http://www.1point3acres.com/bbs/thread-301755-1-1.html>

两轮面试各45min，第一个是个妹子，还晚了5min给我打电话让我以为她要鸽了我。题就是二叉树，第一问问从root开始一条path最大和的值，第二问是leaf-to-leaf 的 path最大和的值，看起来挺套路的不过我没怎么刷过题不知道是不是原题，挺好写的但是有些edge case我第一次写完没考虑到，还有所有的assumption一定要提前说啊！！！我可开心的写完了被一句“what about negative number?” 弄蒙了...我本来以为的是assume 只有非负整数但是我说成了只有整数(。还好有惊无险还都写出来了。不过这个小姐姐巨高冷我面完了都不知道她叫啥。

```
int maxLeafToLeafPathSum (TreeNode* root) {  
    int res = INT_MIN;  
    find(root, res);  
    return res;  
}  
  
int find(TreeNode* root, int& res) {  
    if(!root) return 0;  
    if(!root->left && !root->right) return root->val;//leaf  
    int l = find(root->left, res);  
    int r = find(root->right, res);  
  
    if(root->left && root->right) {  
        res = max(res, l + r + root->val);  
        return max(l, r) + root->val;  
    }  
    return !root->left ? r + root->val : l + root->val;  
}
```

结果因为她来晚了导致我放下电话就是下一个电话，这次刺激了，居然是google research做computer vision的人？？？有可能是因为我resume写得比较奇怪吧.....感觉他们真的按照resume选了interviewer。题目就是design，设计一个窗口界面的class，然后写一个function return 给定一个点的坐标，最上层的window。我用recursion写完了结果他问我可不可以不要递归，我懵逼后他说想一想遍历BST，然后我非常诚实地告诉他我真的没学过递归以外遍历BST的方法，他说那好吧我给你一个新题。后来一查才发现是stack...这个我真的没学，你要说BFS可以用queue这样的例子我大概能想到？新题就是给定一些points ( 带坐标 )，判断这些points是不是完全上下对称的。这道题倒是挺简单的只要写个hashset 确定所有的 fliped point 都在这个array里就可以了。好吧我要吐槽果然做cv的人题都比较古怪.....最后他高冷无比地告诉我啊google research不招本科生intern的，然后介绍了他现在在做的project。

<http://www.1point3acres.com/bbs/thread-300132-1-1.html>

第一轮估计美国小哥吧，一道题，给俩字符串，问其中一个里面交换两个字母的位置能不能得到另外一个，其实做的不好，错漏百出（有的错漏面试官自己也没意识到，他也跟着错），有

follow up，还好沟通无尿点，面试也在吐槽和尬聊中愉快地结束了；5分钟后第二轮，三哥，心凉了一半，沟通有尿点，多次听不清对方说什么。。。也是一道题，题目类似LC 329，做得还好，没

有很注意细节，经三哥提醒也改了，然后问了问时间空间复杂度，问问题部分我了解了一下三哥的狗家生活，问了问team之类的，反正也听不懂，全程awesome，good就尬聊完了

<http://www.1point3acres.com/bbs/thread-303003-1-1.html>

电面第一轮问的是lc 284，然后一堆compare algorithm的follow up

第二轮是输入一堆同义词和一堆句子pair，然后输出每对句子是不是相等，又是一堆follow up

<http://www.1point3acres.com/bbs/thread-304508-1-1.html>

发面经攒人品下午的电面1.2轮是因为...第一轮开始前，google doc当掉了...然后跟小哥各自自我介绍完问完问题尬聊15分钟就...挂电话reschedule了LOL

第二轮直接写题判断一matrix是不是toeplitz matrix，小哥贴了段wiki对这个matrix的定义，就是所有左上到右下对角线的数字都相等就行

具体定义看[https://en.wikipedia.org/wiki/Toeplitz\\_matrix](https://en.wikipedia.org/wiki/Toeplitz_matrix)

**check if [0, n - 2] in first row == [1, n - 1] in second row**

我直接写了递归，对sub matrix的index处理一开始写的有点乱...觉得要挂在这里了...

终于写完follow up是如果matrix很大，一次只能存两行，应该怎么判断

给了三个function: matrix.next, matrix.has\_next, 和matrix.size, 用这三个function写

看到这个我就觉得一开始不应该给自己挖坑用递归的，明明一次读两行更简单...

很快写完最后十分钟问问题

补充内容(2017-11-18 06:41):

一面补面完了.....3sum.....

只要输出true/false.....没有followup.....

小哥是不是想回去放假过感恩节没心情面我.....

<http://www.1point3acres.com/bbs/thread-224642-1-1.html>

LZ程序媛一枚，1.3 Google电面

一面是个印度妹子，感觉人很温柔腼腆。上来就做题。

1. Leetcode house rob3变种，前边说了一堆情景设置，什么员工写report、开party之类的，听得我又迷又懵逼，直到妹子画了BST才反应过来是LC那道题。然后就差不多写好了，问了问时间复杂度。

2. find minimum positive number。我先跟她讲了sort然后brute force的方法，后来正在想有没有更好的方法结果没时间了。妹子还安慰我说因为第一题写的快才出第二题的。。总之这题最后大概讲了下思路，也讲得不太好吧。。

**LC 41**

二面国人妹子，又是个温柔腼腆的妹子，先让我自我介绍一下，然后问题。有很多candidates，他们的votes上有对应的timestamp，首先求出the most vote getter。我是先写出class定义，然后用hashmap做的。.

follow up是求出前k个vote getter，我直接用max heap

follow up2忘了具体的题，好像是和时间有关，反正没写出也没时间了。。

总体感觉差不多跪了吧。。今早却收到通知说进pool了，简直RP爆发。。现在只求team match成功！

<http://www.1point3acres.com/bbs/thread-215552-1-1.html>

一面：

简历瞎说了两三分钟，面试官也无所谓，就直接做题。

1.纯高中计算题：现在有4个广告位置（每个广告位只能放一个广告）。有六个公司想投广告，每个公司有两个想投的广告。问有多少种广告投放方法，使得4个广告位都用完，并且里面至少有两个广告来自同一个公司。

答：如果有4个广告来自三个公司： $C_6^1 \cdot C_5^2 \cdot 2^2 = 240$ ，如果有4个广告来自两个公司  $C_6^2 = 15$ ，最后结果255。

2. 一个多叉树的level traversal，先打印最后一行，每一行先打印最右边的。写了BFS，问了DFS思路没有写。分析复杂度， testcase。

然后大概三十分钟，面试官就说没题了，应该没写出bug这轮。

二面。

小哥介绍了下自己，然后直接跳过了简历。

1.leetcode flip game1, 差别是把翻转加号改成翻转减号。

2.leetcode flip game2, 差别是如果无法make a move是自己赢，不是对方赢。

写了recusive暴力解，分析复杂度说是exponential。写出了bug，逻辑上的bug，面试官指出，然后改正了。

怎么优化，写了hashmap记录中间结果。

还能怎么优化，用dp(不用写，其实我也不知道咋写)。

怎么存每个state更好，我说可以用bit来存，每一位1就是+，0就是-。

又问其实有些string是属于同一个thing（不懂当时），给我举例子：---++ 和--++-++++ 在游戏中实际上是同一个thing（开始还是不懂，后来理解了），怎么判断。

后来发现---++有一组连续两个- 和一组连续三个-；--++-++++也有有一组连续两个- 和一组连续三个-。所以这个两个在游戏中是同一个thing（因为是把--翻转成++）。

<http://www.1point3acres.com/bbs/thread-303132-1-1.html>

第一轮，国人小哥，一道题就是拿到同义词 pair那个，题很简单，follow up 有点难，我用union find 做的，小哥说可以第二轮，三哥，各种听不清，先问了五到关于c++的基本知识，第一道题很简单，就是一个parse string 但是他说data 特别多，所以要拿stream 读，用getline就行。第二题，设计一个画画板，每个pixel 只有1bit, 设计一个数据结构，可以横着画，竖着画。难点就在于每个pixel 是一个bit. 我用int 尺寸的所以一个int 是32个 pixel。

```
void draw_line(char screen [],
               std::size_t width,
               std::size_t x_1,
               std::size_t x_2,
               std::size_t y)
{
    if (x_1 == x_2) return;

    // First byte of the row
    const std::size_t first_byte = width * y;

    // The byte at which we'll start the line
    std::size_t lower_byte = first_byte + (x_1 / 8);
```

```

// The byte at which we'll end the line
std::size_t upper_byte = first_byte + (x_2 / 8);

// The bit in the lower byte
x_1 %= 8;

// The bit in the upper byte
x_2 %= 8;

// Special case when the bits are in the same byte
if (lower_byte == upper_byte)
{
    // Create a mask with the appropriate number of bits
    // and shift them into place. Note that x_2 is exclusive.
    screen[lower_byte] = ((1 << (x_2 - x_1)) - 1) << (8 - x_2);
}

else
{
    screen[lower_byte] = 0xFF >> x_1;

    screen[upper_byte] = 0xFF << (8 - x_2);

    for (++lower_byte; lower_byte < upper_byte; ++lower_byte)
    {
        screen[lower_byte] = 0xFF;
    }
}
}

```

<http://www.1point3acres.com/bbs/thread-299242-1-1.html>

LC 399

google码农电面，我专业数学，一来就问我class，这个会，问我继承Inheritance，会，问我其他的呢？我记得是多态来着，忘了叫polymorphism了，然后问我线程会不会，我说这是啥？问我process是什么？我说我不会然后问我tree会不会，二叉排序数会不会，插入怎么写，然后就是自己提问的时间了。

第二个电话是问我ratio，给一组

A, B, 0.5(A/B=0.5),

C, D, 1.2(C/D=1.2),

...

给两个字符串x, y, 问x/y= ?

然后还有就是可以随时修改 a, b的ratio.

我说把他们看成一个图，找到从x到y的路径，然后x/y就是路径里面ratio的积。

后来他问有没有办法可以避免重复计算，我没想出来，他就说没关系，别想了，就这样吧，bye

现在想想，用个二维数组，无视掉可以随时修改ratio那个性质的话还是可以说点什么的，好气呀，一个这个open的题硬是没说出什么。这个interview半小时就结束了

<http://www.1point3acres.com/bbs/thread-301948-1-1.html>

google research 实习，两轮面试，一轮纯聊research，一轮coding。coding问的题是算汇率转换的题，类似于利口三舅舅。

USD, GBP, 0.69

Meter, Yard, 1.09

YEN, EUR, 0.0077

GBP, YEN, 167.75

EUR, USD, 1.44

GBP, EUR, 1.05

...

YEN - GBP - USD

YEN - EUR - GBP - USD

find\_ratio(10, 'USD', 'YEN')

research的那一轮答的不好，面挂了。但是coding的那道题最近在地里看到过。

<http://www.1point3acres.com/bbs/thread-299169-1-1.html>

第一轮 三哥

没有闲聊直接题目

1. 字符串 删指定字母

2. 给定一个单词前缀和一个字典序排序的字典，找到第一个前缀不符合条件的单词

sorted -> binary search

```
int findfirstUnmatchPrefix(string prefix, vector<string> dict) {
    if(dict.empty()) return -1;
    int plen = prefix.size(), n = dict.size();
    if(dict[0].substr(plen) != prefix) return 0;
    int l = 0, r = n -1;
    while(l < r) {
        int mid = l + (r - l) / 2;
        if(dict[mid].substr(plen) == prefix) l = mid + 1;
        else r = mid;
    }
    if(dict[l].substr(plen) == prefix) return -1;
    return l;
}
```

第二轮 美帝姐姐

依然没有闲聊直接题目

## 1. 判断2叉树是否对称

```
bool isSymmetric(TreeNode* root) {
    if(!root) return true;
    return isMirror(root, root);
}
bool isMirror(TreeNode* t1, TreeNode* t2) {
    if(!t1 && !t2) return true;
    if(!t1 || !t2) return false;
    if(t1->val == t2->val){
        return isMirror(t1->left, t2->right) && isMirror(t1->right, t2->left);
    }
    return false;
}
```

## 2. 判断任意一棵树是否对称

```
bool isSymmetric(TreeNode* t1, TreeNode* t2) {
    if(!t1 && !t2) return true;
    if(!t1 || !t2) return false;
    int i = 0, j = t2->children.size()-1;
    if(t1->val == t2->val){
        while(i < t1->children.size() && j >= 0) {
            if(!isSymmetric(t1->children[i], t2->children[j])) break;
            ++i, --j;
        }
        if(i < t1->children.size() || j >= 0) return false;
        else return true;
    } else return false;
}
```

和三哥的电话杂音很大，交流无比难受，第二题一开始还想错了防线，最后代码细节还没完全写完、第二轮就很顺利做完了。

补充内容 (2017-10-23 03:18):

第二题比如 dict = {"aab", "ab", "abb", "abc", "abx"}, prefix = "aa", return "ab". 如果prefix = "ab", return aab

<http://www.1point3acres.com/bbs/thread-226594-1-1.html>

Iz上周面的，题目可以说算是版上见到过最简单的题目之一。。因为之前在版上看到的题目都各种难，所以发发简单的题鼓励一下像我这样的后进同学。。(Ph.D)

## 1 ) 中国gg

没聊简历，先问问常见sort算法的复杂度。quicksort worst case什么时候发生。问了三道题，第一个是给一个字符串，找出里面所有不是按照alphabetical order出现的字符，很简单，注意统一成lower case就好。第二个是leetcode原题，plus one...第三个是将两个用linkedlist存的number相加，也是原题。。然后时间就到了。每个题都要自己给出几个test cases.

## 2) 印度gg

没聊简历。先问了一道跟中国gg一样的问题，然后我告诉他问过了。就问了一道找出一个string里所有unique character。。。然后问了一道encoding类型的，52张扑克牌，随机shuffle，如何encode shuffle后的信息。最少可以用多少bit来encode。好多地方英语没听明白，有几次答对了也用很质疑的语气说i don't follow you，让我以为自己答错了，浪费很多时间。

<http://www.1point3acres.com/bbs/thread-225315-1-1.html>

binary tree的最长连续序列，秒之。后问了个数列的，具体也记不清了。今天发邮件问结果，说要加面。。。哎。。

LC Binary Tree Longest Consecutive Sequence

LC 549. Binary Tree Longest Consecutive Sequence II

LC 562 Longest Line of Consecutive One in Matrix

<http://www.1point3acres.com/bbs/thread-294735-1-1.html>

1st round :

1. Pascal's triangle，输入两个数行和列，输出对应数值，问了复杂度

2. 如果这个方法被两个for循环连续call，复杂度如何，如何改进，我是用一个List<List<>>全局变量存下计算过的数

**If long long is ok, use  $C(r, c) = r! / c!(r - c)!$**

**if not, LC 118 to compute**

2st round :

1. 判断一个string a是不是通过swap另一个string b中的两个字符得来的，比如abcde 和abedc是符合的，返回true or false  
2. 到这里为止我都觉得一切还ok，然后第二题，说如果这两个string巨长，一台机器完成不了，给你一个List<Workers>，怎么设计，表示对这类问题毫无准备啊，只能硬着头皮上了，定义api就搞了很久，最后也有不少bug，妥妥挂了  
**worker返回一个vector<vector<char>> diffs。最后master汇总workers的这些diffs，当且仅当一共2对 且对换一下char相等，来判断是否是swap得出。**

<http://www.1point3acres.com/bbs/thread-259568-1-1.html>

店面1：<https://programmingpraxis.com/2012/01/20/knights-on-a-keypad/> 可以用DFS解，然后memoization 或者DP优化

```
int knightsOnAKeypad(int n) {
    vector<int> dp(10, 0);
    dp[1] = 1;
    for(int i = 2; i <= n; ++i) {
        vector<int> next(10, 0);
        for(int j = 0; j < 10; ++j) {
            for(int p: m[j]) {
                next[p] += dp[j];
            }
        }
        swap(dp, next);
    }
}
```

```

    }
    int sum = 0;
    for(int i: dp) sum += i;
    return sum;
}

```

电面2:

给一系列近义词的pair, pair间具有传递性 ( 比方说(A, B) (B, C) 可以推出 (A, C)). 然后要求以适当的方式存储pair之后 处理询问类似(X, Y)的pair是否是近义词pair的query.

我当时用HashMap做了， 不过出了点问题， 改了挺久

### LC Sentence Similarity

店面3:

给一个返回值依次为[A, C, C, C, D, D]的iterator, 要求返回另一个iterator, 这个新的iterator要能依次输出如下结果. 鐸櫻鑾宠 [(A, 1), (C, 3), (D, 2)]

我当时一开始没想好怎么做， 边写边想， 结果写得磕磕绊绊的， 估计是因此挂了。 建议 写代码前先把思路想清楚， 不过实在有些细节一下子想不清楚的可以放一放。

后来我又想了一下， 感觉这个题得自己加个buffer， 跟read4k for multiple times有点像

```

class Encode {
public:
    Encode(ArrayIterator it): it(it){
        prev = '$';
        prevCnt = (it.hasNext())? 1 : -1;
    }
    bool hasNext(){
        return prevCnt > 0;
    }
    pair<char, int> next(){
        if(prev == '$') prev = it.next();
        while(it.hasNext()) {
            char t = it.next();
            if(t != prev) {
                pair<char, int> res;
                res.first = prev;
                res.second = prevCnt;
                prev = t, prevCnt = 1;
                return res;
            } else {
                prevCnt++;
            }
        }
        pair<char, int> res;

```

```

        res.first = prev;
        res.second = prevCnt;
        prevCnt = -1;
        return res;
    }
private:
    ArrayIterator it;
    char prev, cur;
    int prevCnt, curCnt;
};

```

<http://www.1point3acres.com/bbs/thread-286100-1-1.html>

要求实现一个计时的 map 和其两个功能，set和get：

set(key, val, time)插入val和time for key

get(key,time) return val if not found return

每个key可以有好几个val和time。我选了dict来实现。因为我用的python，用了defaultdict(list)，面试官一开始没反应过来

tricky部分是，set调用多次，不一定time递增调用，然后get，如果time = 1,3,4 for key1，get(key1, 2), 要求返回time = 1的val。我一开始忽略了，结果直接返回none，因为2不在1,3,4里面。。。面试官（安慰我？）说好多人也犯过这个错误。连连道歉然后猛改。。。越改越紧张，磕磕绊绊。问get的复杂度，O(N)。面试官全程很nice很照顾，就是自己太紧张了！！！

follow up如何优化get，我说keep a sorted list based on time，这样get是logN，不过set也是logN了。

然后就聊天，聊google的开放文化

set: O(1) \* O(logN) = O(logN) in average

get: O(logN) lower\_bound

Lower\_bound upper\_bound 访问到边界最多就是begin()和end(), 所以虽然key < map里的所有值，lower\_bound还是会返回begin()

```

class TimeMap{
public:
    TimeMap() {}
    bool set(int key, int val, int time) {
        timeMap[key][time] = val;
        return true;
    }
    int get(int key, int time) {
        if(timeMap.count(key) == 0) return -1;
        auto& keyMap = timeMap[key];
        auto it = keyMap.lower_bound(time);
        if(it == keyMap.end()) return keyMap.rbegin()->second;
        else {
            if(it == keyMap.begin() && it->first == time) return it->second;
            else if(it == keyMap.begin() && it->first != time) return -1;
            else {

```

```

        if(it->first == time) return it->second;
        else {
            --it;
            return it->second;
        }
    }
}
private:
    unordered_map<int, map<int, int>> timeMap;
};

```

<http://www.1point3acres.com/bbs/thread-289516-1-1.html>

1。 LC 329. Longest Increasing Path in a Matrix

2。 给你一堆横着的竖着的线条，自己定义data structure，找出可以组成的大面积长方形。

(DP and O<sup>3</sup>)

假设随意给你一些横线和竖线，每条线可以定义为 <Point1, Point2>. 每个点就是xy坐标啦。

知道这样一些横竖线，问可以相交而成的最大的面积是多少？面试官要求N<sup>3</sup>算法，我的想法是对于每一条横线，存储它与竖线的交点，然后对于每一对横线，用two pointer，iterate 这两条横线各自与竖线们的交点，左右两边一扫就找到intersection，然后再乘起来，在算一下max product 就好

```

struct Line {
    int c;      // constant coordinate
    pair<int, int> range; // start, end coordinates (start < end)
};

bool inRange(int x, pair<int,int> range) {
    return (x-range.first)*(x-range.second) <= 0;
}

pair<int, int> overlapRange(Line a, Line b) {
    return make_pair(max(a.range.first,b.range.first), min(a.range.second,b.range.second));
}

int maxArea(vector<Line>& hlines, vector<Line>& vlines) {
    int maxA = 0;
    for (int i = 0; i < hlines.size()-1; ++i) // O(N)
        for (int j = i+1; j < hlines.size(); ++j) { // (N)
            auto range = overlapRange(hlines[i], hlines[j]);
            if (range.first >= range.second) continue;
            int xmin = INT_MAX, xmax = INT_MIN;
            int yi = hlines[i].c, yj = hlines[j].c;
            for (auto& vline : vlines) { // (M)
                if (inRange(vline.c, range) && inRange(yi, vline.range)
                    && inRange(yj, vline.range)) {
                    xmin = min(xmin, vline.c);
                    xmax = max(xmax, vline.c);
                }
            }
            maxA = max(maxA, (xmax-xmin)*(yj-yi));
        }
}

```

```

    }
    if (xmin < xmax) maxA = max(maxA, (xmax-xmin)*abs(yi-yj));
}
return maxA;
}

```

<http://www.1point3acres.com/bbs/thread-207690-1-1.html>

1. 给个int array，找它subarray ( 中间indexes连续的那种 ) 里满足  $\min(\text{array}) * 2 > \max(\text{array})$  的那些，return最长的subarray ( 的长度 )。

```

int longest_subarray(vector<int> &nums){
    int max_len = 0;
    int n = nums.size();
    for (int i = 0; i < n; i++) {
        int min_num = nums[i];
        int max_num = nums[i];

        for (int j = i + 1; j < n; j++) {
            min_num = min(min_num, nums[j]);
            max_num = max(max_num, nums[j]);

            if (min_num * 2 > max_num)
                max_len = max(max_len, j-i+1);
        }
    }
    return max_len;
}

```

2. 一些tree的简单问题。。比如，给个tree，还有should\_delete()，删掉某些node 把这个tree 分解成若干个tree，返回各个root。

```

TreeNode *delete_nodes(vector<TreeNode *> &roots, TreeNode *orig_root, TreeNode *root){
    if (!root)
        return NULL;

    root->left = delete_nodes(roots, orig_root, root->left);
    root->right = delete_nodes(roots, orig_root, root->right);

    if (should_delete(root)) {. visit 1point3acres.com for more.
        if (root->left)
            roots.push_back(root->left);
        if (root->right). 1point3acres.com/bbs
            roots.push_back(root->right);
        return NULL;
    }

    if (orig_root == root)
        roots.push_back(root);
}

```

```
    return root;
}
```

第一个是个中国人，一开始闲聊了好久（大概10~15min？）简历上project还有为啥不想读phd啦之类的。。（他说他当年也没有接着读博可能有点共鸣）然后做题。。当时一开始挺紧张的，所以就先brute force了。。recursive那种，写着觉得应该dp，他说是的，然后就加了个hashtable。然后改成了iterative的那种，用2d array存中间结果。后来他说可以改进一下memory，发现其实只需要存O(1)的信息即可。没问问题（没时间了），直接结束。。说实话挺慌的，当时有点小磕绊，还好国人照顾些，有很多hint和交流，人也特别nice。

第二个是个美国人（应该是白人男），他貌似感冒了，全程除了说题几乎没怎么出声，我就一直边码边讲以免场面过冷。。还好题很基础，他也没准备很多问题，最后一看时间还有点干脆让写几个test case。。少有的一些交流也还算顺畅

<http://www.1point3acres.com/bbs/thread-220188-1-1.html>

第一个是一个美国大叔，做infrastructure的，聊了一会儿我的背景和做过的项目。warm up 问了一些算法的时间复杂度，技术题问了 wildcard matching... follow up 让给手动过了几个例子。然后完了还有十几分钟，又聊了一会儿google cloud。

第二个是一个国人或者(ABC)小哥，一开始聊背景，做的research，然后问了 full binary tree（每个node 要么有两个 child要么没有）的问题，求给定leaf node数量，生成所有可能的树的结构。做完还有点时间，他介绍了一下他intern 和 fulltime的经历，他对google的看法。

```
public static int fullBinaryTreeNumber(int n) {
    int[] dp = new int[n + 1];
    dp[0] = 1;
    dp[1] = 1;

    for (int i = 2; i <= n; i++) {
        int count = 0;
        for (int k = 1; k < i; k++) {
            count += dp[k] * dp[i - k];
        }
        dp[i] = count;
    }
}. From 1point 3acres bbs
return dp[n];
}
```

```

public class GenerateFullBinaryTree {
    . From 1point 3acres bbs
    public static void main(String[] args) {
        GenerateFullBinaryTree gfbt = new GenerateFullBinaryTree();
        System.out.println(gfbt.generateFullBinaryTree(1).size());
        System.out.println(gfbt.generateFullBinaryTree(2).size());
        System.out.println(gfbt.generateFullBinaryTree(3).size());
        System.out.println(gfbt.generateFullBinaryTree(4).size());
    }

    public List<Node> generateFullBinaryTree(int n) {
        List<Node> res = new ArrayList<>();
        if (n == 0) {
            res.add(null);
            return res;
        }
        if (n == 1) {
            res.add(new Node());
            return res;
        }
        for (int i = 1; i < n; i += 1) {
            List<Node> leftNodes = generateFullBinaryTree(i);
            List<Node> rightNodes = generateFullBinaryTree(n - i);
            for (Node l : leftNodes) {
                for (Node r : rightNodes) {
                    Node root = new Node();
                    root.left = l;
                    root.right = r;
                    res.add(root);
                }
            }
        }
        return res;
    }

    class Node {
        int val;
        Node left = null;
        Node right = null;

        public Node(int v) {

```

```
        this.val = v;
    }
    public Node() {
        this.val = 0;
    }
}.
```

}

<http://www.1point3acres.com/bbs/thread-236030-1-1.html>

第一个是一个印度姐姐，口音很重。只要说长句子，就听不懂在说啥。不过考的不难。考了两题，一个是invert binary tree, 一个是给你两个数组，让你找出在数组一中，但是不在数组二中的元素。第一题就是leetcode上的easy题，第二题用的map..感觉都不难。就是说话听不懂。follow up: 复杂度

第二个是中国姐姐。考了一个有点麻烦的word match 的题。给你一个word dictionary, 例如：`<FaceBook, FootBall, SunGlass, GlassDoor>` 他们对应的缩写为FB , SG, GD...现在给你一个FakeBook让你把对应的FB所有的FaceBook FootBall找出来。如果找不到对应的就返回空。题目大概是这样。记得当时写了google doc满满一页，还翻了页。 follow up: 复杂度

过了几天，通知还要面一轮。估计是印度姐姐那个沟通太困难

这次是个美国小哥，从Mountain View打过来。告诉我他中午做了什么吃，还要我自我介绍了一下。然后开始做题。(感觉美国人就是比较注重缓解气氛 )

考的是Number of Islands的变种。把islands个数找完后。输出最大islands的面积。比如一个islands 是与其它四个islands 连着在，他们最后就会算做一个island, 大小是5. 返回的就是这个5. 记得当时就是在helper function 里面弄了个variable 记住了当前的islands, 如果下一层4个方向dfs的时候也是island, 就 + 1...然后helper function 返回这个variable. 写完后有两个小bug..改了改

Follow up: 如果不想改变原有的二维数组要怎么做？答了三个办法：第一make a copy; 第二，把1的地方变成 - 1，再变回1; 第三，用map,记住元素的位置

<http://www.1point3acres.com/bbs/thread-285486-1-1.html>

第一面是read two files

File 1 :

USD, GBP, ...

Meter, Yard, ...

YEN, EUR, ...

GBP, YEN, ...

File 2:

USD, EUR

Yard, Meter

第一个file中...是汇率数字，我不记得具体多少了用这个表示一下，而且不止这么多，会有很多不同的单位之间的转换。然后要求output一个file，计算第二个file中给出的单位之间从第一个到第二个的汇率数。

naive:

DFS for every queue, but it's too slow if too many queue

Improvement:

convert distance to log

EX: USD -> TWD , 30 → USD -> TWD  $\log(30)$

So that the  $\exp^{\text{distance}}$  is the exchange rate  $\exp(\text{dist}[S, Y] - \text{dist}[S, X])$

第二面是给a list of dices, 每个dice不是用数字表示，而是用字母，相当于给一个list的包含六个各种字母的String，然后再给一个target word，判断这个word是否可以由这些骰子摇出，要求不能两个或两个字母以上由同一个骰子出来，但骰子数量可以很多，骰子上字母可以是任意字母，重复也行。

<http://www.1point3acres.com/bbs/thread-283995-1-1.html>

本科面狗家秋季的intern 6/\*\*日面的

第一轮：一个美国小哥，总体感觉还是心急一点。前面5分钟左右让我自我介绍，题目是：一个青蛙可以跳最多m步，然后要到达n层（步数超过n层不重复计算，就如果最多可以走3步，要到2层，那走3步和走2步算是一种），然后return一共是多少种走法。上手很慌，小哥比较在意思路，要先和他讨论。然后我一开始说打算用dp做，我说我觉得到达k ( $m < k < n$ ) 层的方法就是到k-1层的方法加上k-2层的方法加上k-3层...一直加到到k-m层的方法。但是我当时在纠结如果k小于m的时候咋算，当时确实紧张了些所以想了半天。小哥说你觉得你怎么样去想？我说我找几个example自己找找规律，他说可以。反正后来搞明白了开始写，发现还是要结合递归去做，然后花了些时间到最后40分钟的时候左右差一点快写完了，他说他要走了没时间了（说好是45分钟的，他确实早走了）。小哥说思路是对的，差不多对了，可惜没时间给我问问题了。我内心很崩溃，我觉得题目不难，就是自己水平不足。

```
int frog(int n, int m) {
    if(n == 0 || m == 0) return 0;
    vector<int> dp(n+1, 0);
    dp[0] = 1;

    for(int i = 1; i <= n; ++i) {
        for(int j = 1; j <= min(m, i); ++j) {
            dp[i] += dp[i - j];
        }
    }
    return dp[n];
}
```

第二轮：一个国人姐姐，3分钟自我介绍。然后出题lc297，serialize和deserialize，因为之前看过这题所以思路比较明确。姐姐压迫感还是很强的，我依旧很紧张。我说我打算pre order递归建string。然后姐姐给了个二叉树，让我先按照我的思路笔头serialize

7.13日（完成加面）题目是paint fence问你两种颜色漆n片fence，follow up是k种颜色。面试官印度小哥，我表达的不顺畅，题是做出来了。她说ok，然后问了下runtime~

<http://www.1point3acres.com/bbs/thread-285482-1-1.html>

上来一个三哥，问我知道batch和streaming会不会，我知道大事不妙，简单解释了几句。开始coding，LC4直接上手，假设input很长很长，这题昨晚做过所以很开心写完了，然后来了一道read median from stream，这题也准备过，所以coding勉强撑了过去，最后十分钟问了两个stream process的概念问题，当场爆炸，楼主根本不会，互相问好然后挂了，觉得三哥并没有给很难的题，那些概念我确实应该准备一下的，血泪教训

LC 4 Median of Two Sorted Arrays

LC 295 Find Median from Data Stream

休息了一刻钟第二轮，声音很好听的白人姐姐，在youtube干活，最大的爱好是看youtube。。。。。上来问了简单的概念，stack queue deque是什么，java用什么实现deque，arraydeque和linkedlist大概是什么，有啥区别，这个糊弄过去了，然后让我写一个generic array来实现deque，楼主差点想挂电话，根本没写过generic，于是就用int[]大概实现了，因为之前没写过，所以临时想了array的长度不可变，所以有时候需要替换新的更长array，poll时我定义了start和end，这样不用删除时再换新的array，姐姐全程一直“very good”，让我觉得她会放水，好不容易写完了她开始问generic的问题。。。。。然而楼主并不懂，很尴尬，最后给我五分钟问问题，我说youtube广告为啥有的不能skip，姐姐说是啊好讨厌啊影响姐看视频

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312707&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

第一轮：将一个函数运用到列表中的每一个元素上，有点像map

问了问上面这个返回结果如果用cache存储，空间是多少，工业上应该如何设计，查找一个元素时间复杂度是多少，BST和hashmap各有哪些优势，hashmap的空间是多少

给一个number，在不使用内建函数的情况下转换成相应字符串。例如：10001->'10001'

```
template<typename T>
void myMap(vector<T> &vec, function<T(T)> func) {
    for(int i = 0; i < vec.size(); i++){
        vec[i] = func(vec[i]);
    }
}

int main() {
    vector<int> a = {1,2,3};
    function<int(int)> func = [](int x) {return x + 1;};
    myMap(a, func);
}
```

第二轮：如何在一个binary tree中找到extra edge并断开它，follow-up如果有多条边怎么办，一样的

第三轮：

给你一个pattern ( pattern中digit表示匹配相应个letter，letter表示匹配letter本身)，一个string判断是否match：

ex:

abc->'abc'

'1oc3'->'aoczzz', 'bocabc'

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314354&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

第一轮：

给一个直角坐标系，系上给若干点，求任意两点，满足：这两点的连线将剩余点 on one side of this line, 就是说这条线可以把剩余点分到线的一边，而不存在这条线的两边都有点的情况。

一看题目很懵，但是面试官一直给hint. 先是问，给N个点可以画多少条直线？秒答  $N*(N-1)/2$ ;

对了， struct P{double x;double y};

然后给了一个API，说 int OnWhichSide(vector<Point> line, Point p); return -1表示在一遍，return 1 表示在另一边。

题目的function signature是(这是我自己的写的) vector<Point> PointsOnOneSide(vector<Point> input)

```
pair<Node, Node> foo(vector<Node> nodes) {
    Node leftMost = Node(numeric_limits<double>::infinity(), 0);
    for(auto node : nodes) {
        if(node.x < leftMost) {
            leftMost = node;
        }
    }

    for(auto node : nodes) {
        if(node == leftMost) continue;
        auto line = getLine(node, leftMost);
        int sum = 0;
        for(auto other : nodes) {
            if(other == leftMost || other == node) continue;
            sum += onOneSide(line, other);
        }
        if(abs(sum) == nodes.size() - 2) {
            return make_pair(leftMost, node);
        }
    }

    Node error = Node(-1, -1);
    return make_pair(error, error);
}
```

于是我就答把所有点遍历一遍，组成line，再判断其他点是否满足条件，如果满足，直接return。

问了复杂度  $O(n^3)$  if int OnWhichSide(vector<Point> line, Point p) has  $O(1)$ . 这题就这样结束了，全程面试官很温柔，一直good good，有bug也是让我go through一遍function自己发现的。

第二轮：

第二轮有两道题，感觉面试官没有上一个Nice，我心态就有点崩，仔细看下来觉得挺简单的，但是全程答得颤颤巍巍。

说有一个struct Message{string source\_file;string content;}; 然后给 vector<Message> v =

{["a","loop1"], ["b","loop1"], ["c","loop1"]}, {"a","loop2"}, {"b","loop2"}, {"c","loop2"}, {"a","loop3"}, {"b","loop3"}, {"c","loop3"}, {"a","loop4"}} 类似的vector。说要return一个新的 unordered\_map<string, vector<Message>> such that the map is like  
{{"a", [{"a", "loop1"}, {"a", "loop2"}, {"a", "loop3"}]}, {"b", [{"b", "loop1"}, {"b", "loop2"}, {"b", "loop3"}]}, {"c", [{"c", "loop1"}, {"c", "loop2"}, {"c", "loop3"}]}};

第二题是vector<Message> truncate(vector<Message> v,int num);

就是说num是多少，这个return的vector最多含有num个相同的source file. 比如对于上面那个例子，如果num是2，那么return的就是 {"a","loop1"}, {"b","loop1"}, {"c","loop1"}, {"a","loop2"}, {"b","loop2"}, {"c","loop2"} (最多2个a，2个b，2个c)。

最后发点牢骚，从12月13号开始，收到google 和 facebook 的interview，觉得自己google无望，转头去准备fb的面经和tag题，刷了2遍吧at least，最后被fb当天就拒了，google完全抱着不可能，很放松去面的，结果给了加面（说明有戏？加面到底是什么情况。。。），有点郁闷，等心情好一点再发fb跪经吧。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314207&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

第一轮是个刚工作的印度小哥，问给一个数组A，再给一个shuffle A得到的数组B，再给个新数组C，求shuffle C得到的数组D。要求C到D的映射与A到B要一致。假设所有数组都未排序。

naive的解法很简单，建立一个散列表存A到B的映射关系搞定。但是时间复杂度是n平方，因为B中每个元素都要花线性时间在A中查找。

```
unordered_map<int, vector<int>> dictB;
```

```
for(int i = 0; i < B.size(); i++) {
    dictB[B[i]] = i;
}
for(int i = 0; i < A.size(); i++) {
    dictA[A[i]] = dictB[B[i]];
}
```

Follow up: 1. 如果数组中有重复元素怎么办？2. 如何优化时间？

1. 很简单，关键在于2. 如果题设给定的A和C是排序过的自然可以用二分法，这样时间可以降为nlogn。然而问了小哥说没有排序过，于是想到A排序后变为A'，再用A'到B，分别存A-A'，A'-B 的两套映射，再如法炮制C和D。但是写到一半发现这个方法也不对。。。尴尬之余小哥提醒时间快到了，于是进入向他提问的环节。。。事后我还是想不出有什么好的优化方法，我估计小哥自己也不知道吧他好像也没有太care最后的优化结果。

小哥人很好，一直给我反馈，我有问题回答的也很快。

第二轮感觉是个白人小哥，问的是之前面经原题，给一个长方形框（宽和高）和一段话，问能把整段话放进框里的最大字号，你可以调用两个现有的函数，分别返回每个字符的宽度和字符高度。

```
check(string s, int size) {
    Int h = height / charHeight(size);
    Line_pos = 0;
    Int curr_height = 1;
    for(auto c : s) {
        Int w = charWidth(size, c);
        line_pos += w;
        if(line_pos > width) {
```

```

        curr_height++;
        line_pos = w;
    }
    if(curr_height > h) return false;
}
return true;
}

```

典型的二分查找，在字号范围内不断二分，最后就返回能把整段话放进框里的最大字号了。

这轮看到了原题，窃喜之下心态没绷住，背了一波网上看到的解法。。。没想到那个解法的check function有问题，加上自己心态太急没有仔细思考，被指出一个严重的疏漏，一慌之下重写又连出bug。。。这轮估计主要就是挂在这上面了。所以教训1 ) 心态要稳 2 ) 网上的解法很多有误，只有自己写一遍才是自己的。

之后约了8号的加面，第三轮是个白人妹子，听声音感觉很活泼可爱那种，给一个0和1的矩阵，其中0是墙，1是路。问有没有路径从第一行（任意位置）到末尾行（任意位置）。

DFS不多说，很简单。

Follow up：同样的矩阵，返回从第一行（任意位置）到末尾行（任意位置）的最短路径长度。

BFS不多说，也很简单。

吸取了前两轮的教训这轮花了更多时间检查bug，确实检查出了一些bug，避免了好几个坑。

不太懂为啥这一轮只有45分钟，妹子说本来还有个问题的现在时间不够了，然后我尬问了几个问题就结束了。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313616&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

十二月初面了背靠背的电面，两轮连着各45min，都只问了我一道题~

第一面是白人boy，先问了我一些简历上的问题，讲了快十分钟才开始做题的。题目是假设有一串链表：随机输入其中的几个node，让你判断一共有几个连在一起的区域

比如说这串链表是这个样子的 a->b->c->d->e->f->g->h，输入序列是c, d, e, a, h, g，因为c d e是连在一起的，所以可以看做一块区域，同理g h也是连在一起的，所以应该输出3

题目看懂之后挺简单的，定义一个set<pair<\*node, \*node>> 就很好做了

第二面是印度girl，感觉有一点点奇怪，因为并没有让我coding，只是让我设计一个存储数据的结构，再不断给我提出不同的假设改进这个结构。

每个数据都是一个key-value pair，还有一个expiration time，看怎么高效地设计和提取这些数据

面完过了一周收到了加面的信息，约到了昨天。加面是一个白人girl，人很好，上来直接做题。题目也非常简单，输入是一个文件，其中每一行都是 timestamp <username> content 这样的。要求输出 top N users who have the biggest word count。我先用hashmap来计算叠加每个username的word count，然后用max heap来求前N大的。需要自己写比较函数，

因为要保证在word count一样的时候不同username在max heap里的大小关系是确定的。我已开始粗心，把word count直接看成了content.length()，经面试官提醒才意识到这个要求word的个数。也很简单，仔细就好。.  
大概就是这样啦~求人品啊求人品，顺便走过路过都求一下大米！！！！

//拓扑排序, thread, 子树结构，676，union find vs dfs, 平衡union find, tape存数据 put(key, value, duration), get(key)  
, lc399, kmp重复子串, 转账最小次数, segment tree, swap n次得到另一个string  
<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313973&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

今天下午刚面完的谷歌intern店面，两轮45分钟背靠

1. 中东小哥，input两个text文件，第一个叫data, 每行格式是unit\_a, unit\_b, exchange\_ratio, 例如USD, CNY, 6.8或者MILE, KM, 1.6等。 第二个叫query, 每行格式是unit\_a, unit\_b

要求在query文件每行的末尾加上对应的exchange\_ratio。做完闲聊发现小哥是Caltech的phd，认识我老板，我还引用过他的paper，缘分啊。。。

2. 国籍不明说话有回声大叔，题目是一群人一起去吃饭，买单的时候每人出的钱不一样，回家以后他们各自用微信相互转账，使每个人出的饭钱相等。input是一个int array表示买单的时候每个人出的钱，要求output转账的时候谁给了谁多少钱，比如 B->A \$3, C->A \$1。大叔是infrastructure组处理网络异常的，不过g的面试官跟以后分组好像没啥关系。两轮都没有简历问题没有bg，都是做题+闲聊

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=310765&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

在国内用hangouts面的，早晨6点爬起来搞，心累。信号稳定毫无问题

一面小哥没有口音声音好听。

给一个string，可以在任意部分插任意个char，使得其为回文，求最短的回文

毫无思路，被小哥一路提示加暗示做出来了，复杂度爆了，follow up就是加个memo，然后提出几个bug我改了，然后一些代码优化，没有优化完就结束了，可以说是雪崩

蠡口尔妖寺的豪华升级版

```
void LCS(vector<int> &idx1, vector<int> idx2, int &commonLen, string s1, string s2) {  
    vector<vector<int>> dp(s1.size() + 1, vector<int>(s2.size() + 1, 0));  
  
    for(int i = 1; i <= s1.size(); i++) {  
        for(int j = 1; j <= s2.size(); j++) {  
            if(s1[i] == s2[j]) {  
                dp[i][j] = dp[i - 1][j - 1] + 1;  
            }  
            else {  
                dp[i][j] = max(dp[i][j - 1], dp[i - 1][j]);  
            }  
        }  
    }  
  
    commonLen = dp[idx1.size()][idx2.size()];
```

```

int i = s1.size(), j = s2.size();
while(i >= 0 && j >= 0) {
    if(dp[i - 1][j] == dp[i][j - 1] && dp[i][j] == dp[i][j - 1]) {
        i--;
        j--;
        idx1.push_back(i - 1);
        idx2.push_back(j - 1);
    }
    else if(dp[i - 1][j] == dp[i][j]) {
        i--;
    }
    else {
        j--;
    }
}
}

//the len, number of char needed to be add = len - n * 2;

string formPalindrome(vector<int> &idx1, vector<int> &idx2, const string &s1, const string &s2) {
    string res;
    int i = 0, j = 0;
    for(int k = 0; k < idx1.size(); ) {
        if(i < idx1[k]) {
            res += s1[i];
            i++;
        }
        else if(j < idx2[k]) {
            res += s2[j];
            j++;
        }
        else {
            res += s1[i];
            i++;
            j++;
            k++;
        }
    }
}
}

```

二面yd小哥有口音但沟通流畅

1. 给一个 string , 给一个函数isWord ( String ) 检查它是不是个word ; 每次可以remove任意一个char , 然后再检查它是不是word , 直到string长度为1 , return true ; 否则 false ;

```

bool isWord(string s, unordered_set<string> dict) {
    helper(s, dict);
}

```

```

bool helper(string &s, unordered_set<string> &dict) {
    if(dict.find(s) == dict.end()) {
        return false;
    }
    if(s.size() == 1 && dict.find(s) != dict.end()) {
        return true;
    }

    bool res = false
    for(int i = 0; i < s.size(); i++) {
        char tmp = s[i];
        s.erase(i);
        res = res || helper(s, dict);
        s.insert(i, tmp);
    }

    return res;
}

```

example : hat -> at -> a true word 去掉哪个都不是word false

2. 半页的描述，貌似复杂，但是一个二分就秒了

小哥说you do perfect

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313645&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

之前狗的全职电面面的不好，所以进了这个residency program，来了两轮背靠背。.

第一轮是个印度姐姐，让写一个int steps(int n)，返回n到1需要多少步，如果n是奇数  $n = 3n + 1$ ，偶数是  $n = n/2$ ，我先用iterative的方法写的。然后再让写一个 int maxSteps(int N), 输出  $1 \leq n \leq N$ 这个区间内最大的steps(n)，很容易看出redundancy，我就说用Map存一下，这个map是  $n \rightarrow \text{steps}(n)$ ，结果implement的时候卡住了，经姐姐提醒，把steps(n)的方法改成了recursion，这个map是要maxSteps和steps共用的。因为想了一下map放在哪里还有迭代改递归，然后一被challenge就紧张，所以写的慢，感觉面的不太好。

```

int stepsCount(int n, unordered_map<int, int> &dict) {
    if(dict.find(n) != dict.end()) {
        return dict[n];
    }
    if(n == 1) return 0;

    if(n % 2 == 0) return stepsCount(n / 2, dict) + 1;
    else return stepsCount(3n + 1, dict) + 1;
}

```

第二轮是个国人大哥，题目描述不同但是基本上等于 house robber III 多叉树，这题我没刷到，但经大哥提示写出来了。

大哥人特别好。。。,

昨天得知加面，估计被印度姐姐挂了。。人生不易 且行且珍惜。。。发面经攒人品。。。求大米。。。

<http://www.1point3acres.com/bbs/thread-313738-1-1.html>

前面的，两轮题都比较简单，然而没把握好。。

第一轮，网络infra组一个老哥，有点口音，给一个color数组[R,R,R,G,B,B]，随机返回一个color，概率和color出现次数成正比。我直接给了个取 $i = \text{rand}(0, \text{color.length})$ 然后返回 $\text{color}[i]$ 的方法，他说不行因为我们希望把input当成数据流。。。这个问题可以用reservoir sampling做，然而我当时是想了个用count数组的办法，中间卡了一会被老哥给了个大hint然后磕磕绊绊写出来了。follow up要求优化时间，对count数组做prefix sum然后二分可以达到 $O(\log n)$ 。这一题感觉是热身的然而我做完就没时间了，然后问了几个问题就结束了，血崩。。

第二轮，cloud组美国小哥，让我自我介绍，然后问我为什么想做database。尬聊五分钟后出题，给一个数组表示汽车出发顺序和速度，速度快的车最终会困在速度慢的车后面形成cluster，求cluster数量。题不难，遍历数组不断的找下一个慢车就行了。。follow up要加第n+1辆车，速度比其他车都快，出发时间不确定，求加入这辆车以后形成cluster的所有情况。给了个 $O(n^2)$ 的解法。题不难写但是小哥顺着我的代码问了一大堆基础问题，以至于我本来以为能看到第二道题结果还是一道题就结束了。。。

感觉面狗家之前做题真的应该每道题都好好分析一下时间复杂度空间复杂度，想当然的说一个答案常常是错的。

两天后收到加面，应该是第一轮挂了，加面求好运吧。。。

<http://www.1point3acres.com/bbs/thread-309699-1-1.html>

第一轮：想象一个只有许多长度不一的横着的长条方块的俄罗斯方块场景，问所有方块下落后的样子，长条不会倾斜，只要某一个地方搭住下一层就停止下落，输入输出形式自定。感觉考点在怎么定输入输出的形式，以及如何判断一条能否下落到下一层。这轮写的时候面试官一直在问我的做法中可能出现的问题和优化，写得比较断断续续，估计feedback一般

interval方法写，定义一个hasOverlap函数判断当前长条是否和要比较的一层有重合。

vector<vector<Interval>> done//已经放好的长条；

//每次放入新长条

Interval curr;

```
for(int i = curr_height; i >= 0; i--) {  
    if(hasInterval(done[i], curr)) {  
        mergeInterval(done[i], curr);  
    }  
}
```

第二轮：给一个矩阵每个元素是该位置的高度，矩阵的一侧有一条河高度为h，求水漫进矩阵后的结果，bfs秒之。follow up假如没淹掉的地方两个位置有一个人和一条狗，人想要不涉水到达狗的位置，河高度不定，问最大的高度h使得人走的步数小于等于给定的k。遍历高度h，调用第一问函数，bfs求路径长度，复杂度 $O(h * n^2)$ ，没写代码，如果二分搜索h复杂度 $O(\log h * n^2)$

11.6 收到通知加面

11.14 加面

比较简单。第一题2-sum，三种方法秒之。第二题交换左右子树，秒之。第三题ratio，本来一根筋想用map<char, pair<vector<char>, vector<double>>，后来面试官说你用一个class node做吧，秒之  
12.13 通知project selection，目测之前feedback拖了很久没交

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313650&extra=page%3D3%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

体验不好

这个老哥说话很慢，比我慢得多，又含混不清，考托福口语估计得15分

可能是老墨或者拜仁

上来先问我有几个T的一堆string如何排序，我说prefix tree 然后不同prefix放在不同distributed servers上跑sort然后sort好了合起来行不行。他说没懂。.

给老哥讲了半天啥是prefix tree，怎么compare，怎么sort。然后他说懂了，问我具体怎么操作 ( How to do it)。我：？？？然后我就举了两个例子aaaab和aaaac怎么排序。他说你说的我都懂，具体怎么操作 ( How to do it)？我：？？？？

我说你哪里不懂呀，他说这么大数据量怎么建tree。我就解释不同prefix的string扔到不同server上各自sort好了直接拼，不同server上的string不用compare blabla。他说ok fine，估计还是没懂.....此时耗时15min

然后coding，实际上就是LC俄舞亿，但是他给我的是一个调用函数，输入iter of iter 返回 iter。我装不懂的问他，我是不是要implement hasNext和next，他说不太清楚，你就给我个iterator就行。我：？？？啥？

就懵了，我说那我能不能自己定义一个class myIterator，他说不确定，反正你最后给我个iterator就行。我：好吧，那我扯开这个2D iter的数据，用个list储存，然后返回这个list的iterator行不。两行写完。他说looks good如果有很大的数据量呢，list存不下呢。此时耗时30min

我想那不就还是自己写class= =自己定义一个iterator了，刚把框架列上去，他好像很高兴说对对这样great。

然后5min把答案撸出来，时间就到了。

```
bool hasNext() {
    if(it != outer.end()) {
        return true;
    }
    return false;
}

bool next() {
    int res = **it;
    *it++;
    if(*it == inner.end()) {
        it++;
    }
    return res;
}
```

总之这次这个面试官体验很不好，感觉他自己也不是很懂，只是有题和标准答案而已。与之前背靠背的两次都在努力引导我正确方向和提示完全不同。

不知道这种情况会给出什么样的评价。只能说希望能有好结果吧。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314107&extra=page%3D3%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

1. 在一个巨大字符串流中找第一个unique char

solution: 简单题，跟leetcode解法差不多。但是面试官的expectation很高，我基本把我能想到的各种解法(one pass, two pass, hashmap, linkedlist)都写出来了。最后面试官还问我怎么用一个HashMap<Character, Boolean>去做，我当时没做出来，比较tricky。后来面试官又跟我说了下做法，这样的话得two pass，没有我说的optimal solution one pass好，blabla

2. input是一个矩形，生成矩形里的随机点。followup 1: input有很多矩形，不会重叠；followup 2: input很多矩形，而且会重叠。

之前面经也出现过，典型题。followup2没有要求实现，我大概说了下思路，中心思想就是分割成一个个竖矩形。先把矩形分成左右edge，然后edge根据x coord排序，然后左边扫描线扫过去。事前可能还需要做下discretization，把所有坐标对应到不同的integer index；扫描过程中可以运用线段树优化，使每次query和update控制在logN。大家可以参考Atlantis这道POJ的题，是求重叠矩形的总面积，很经典的扫描线+线段树。

3. dp题，类似于path count，不过稍微复杂一些，followup比较繁琐，我就不说了。

4. 生命游戏, lc289, 这题跟leetcode不一样的地方是input 是二维boolean数组。我首先new一个新matrix存放结果；面试官要求降低空间复杂度，不能new matrix；input是boolean不能用来像int那样利用一些冗余bit，后来我搞了个滚动int数组，保存当前行和下一行的count，对于每个cell center给下面三格和右边一格 +1，反过来也+1。这样的好处是cell不会访问之前已经处理过的结果。面试官说that might work.

5. 利扣得起散斯 起散起，注意有些变化。起散起跟题目不一样的地方就是，原题是A->B, A->C能推出B->C，我被问的时候，面试官说不能推出B->C。所以这道题只能保证transitive, 没保证symmetric. 所以这个图是有向图，有向图不能用Union Find做，只能老老实实用DFS.

<http://www.1point3acres.com/bbs/thread-314260-1-1.html>

电面面经：

应该是面试官口述的新题吧。

LZ Compression:

Decompressed: ABRA KEDABRA

Compressed: ABRA KED[0xFE 4 7]

(tag = 0xFE/254, len, offset)

Decompressed: ABRA KEDABRA DABRA

Compress: ABRA KED[0xFE 4 7] [0xFE 5 5]

遇到0xFE说明遇到压缩的部分了，第一个是压缩的长度，第二个是压缩的offset，比如第一个case，从D开始往回offset到A，压缩的string是ABRA

给定compressed string,string中每一个字符都用byte表示，0xFE看作一个。返回decompressed string

input : ABRA KED0xFE 4 7

output:ABRA KEDABRA

<http://www.1point3acres.com/bbs/thread-312698-1-1.html>

LEETCODE 贰零

多线程解决

面试官是个阿三

我简历上根本沒有多线程

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314065&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

题目：给一个input txt file: 里面有video的name和watch time (有重复)，要求1) 输出top ten watch time的video name以及watch time 2) 输出top ten view times的video name及view times。这两问其实异曲同工。最后应该是用heap做就好了，但是一开始死活没想起来，说了priorityQueue.....虽然最后我想起来heap，但是时间到了就简单说了下怎么做，没写code.....姐姐给了我一些建议，嗯，还是挺有用的。面试过程中要think it aloud, 虽然我觉得我已经think it aloud了，但是有时候在思考就会有点quiet.....

同时附上电面一、二的题目吧。

电面一：

第一题：把三进制转换成十进制。

第二题：里口撒斯领。sliding window的解法。

电面二：

就一题，是类似里口那个skyline的题目，给出skyline的矩阵，输出最大的possible的volume。后来followup的时候问如何优化算法什么的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312170&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

问了一道题

给一个数组，n的长度 (  $n \geq 1$  )，含有所有1 - n的数字，要求得到一个最小的index，满足条件：如果把数组小于等于index的部分排序好，并且把数组大于index的部分排序好，那么整个数组就排序好了。

看题目可能有点绕，以下是当时给的两个examples：

3 1 2 5 4

返回的index为2，即如果(3,1,2)和(5,4)分别排序完成，那么整个数组就排序好了。

2 4 3 1. more info on 1point3acres.com

返回的index为3

注意由于含有所有的1-n的数字，所以数组里是没有重复元素的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314184&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

扫地机器人。

Robot Clearner

Given a robot cleaner in a room modeled as a grid. Each cell in the grid can be empty or blocked. The robot cleaner can move forward, turn left or turn right. When it tries to move into a blocked cell, its bumper sensor detects the obstacle and it stays on the current cell.

Thecontrol API:

```
interfaceRobot {  
    //returns true if next cell is open and robot moves into the cell.  
    //returns false if next cell is obstacle and robot stays on the current cell.  
    booleanMove();  
    //Robot will stay on the same cell after calling Turn*. k indicates how  
    //many turns to perform.  
    voidTurnLeft(int k);  
    voidTurnRight(int k);  
    //Clean the current cell.  
    voidClean();  
    booleanMove(Direction d);  
}
```

Sampleinput

```
++++++  
+.....+  
+...^...+  
+.+. ....+  
++++.++++  
+....+  
++++++
```

要求用给定的API打扫完所有empty格子。 Time complexity, linear in term of room space.

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312275&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

1. 王位继承，长子及其后代有优先继承权，然后是次子及其后代，依次类推。完成birth(parentname, childname)和death(name)以及输出王位继承顺序三个function

2. 给一个有向图返回所有从起点到起点的环组成的图，输入只有一个起点

3. 判断在字典中是否存在改动一个字母就可匹配输入字符串的单词，这题时间复杂度没答好

4. 字典中所有单词长度为5而且每个单词都没有重复字母，实现一个function可以生成下一个猜测的词

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313878&extra=page%3D5%26filter%3Dsortid%26sortid%3D311%26sortid%3D311>

一个多小时前刚刚结束的狗电话。楼主16Fall转的计算机，申请的是18年的SDE NewGrad。

一道完整的题，一点点的展开，很狗，很狗。

1. 让我写一个"人"类，含有名字跟年龄。用private+setter/getter，楼主琢磨考encapsulation

2. 让我写一个"码工"类，楼主琢磨考inheritance

3. 让我区分“码工”类里面，各种variables跟methods，是public/protected/private/default和why
4. 让我写一个“给予”跟“接收”的方法
5. 把第四步的方法，多线程化，考multi-threading + synchronization
6. 哪里会锁死和如何解锁（时间到了，就没细问）

作为转专业狗，我很认真的看了里扣高频跟新题，过去几天硬着头皮背了一些算法（插一句话：我真的很佩服那些能够理解并且写出复杂算法的人，楼主只能靠死记硬背，虽然知道这样不好），不想题目竟然是这样。

提醒一点后来人吧：别家问encapsulation和inheritance，是问这个东西什么意思，或者给你一段程序，问这个结果是什么（Mathlab最喜欢考读程序，亲身经历）。狗家一声不知，直接说你写一个类看看，也不说要用encapsulation，是个小坑。当然，也有可能是我道行不够，还有更多的坑我没有发现，后来人慎重，慎重。

<http://www.1point3acres.com/bbs/thread-313876-1-1.html>

native小哥，没问问题直接做题。公司有一堆电梯，设计一个控制系统，优化目标是乘客等待时间最短。面了一个小时，小哥疯狂给hint。最后没问问题就结束了。小哥说太复杂了没关系（你太菜了挂的活该）。

<http://www.1point3acres.com/bbs/thread-310930-1-1.html>

一面：

实现一个会过期的Map。有put(key, value, lastTime)和get(key)两个方法。每个k-v对放入Map时附带一个有效时间lastTime，持续到过期前都可以正常get，过期后再get则返回无效标记。

关于lastTime什么数据结构我当时询问了面试官，他表示他也很用JAVA不是很知道里面用什么时间库，就让我assume一个long getCurrentTime()方法来做了，于是解题中是用long来表示时间。

follow ups: 1.如果你出unit test你会出什么case。2.时间复杂度。3.如果要实现一个cleanUp(timeStamp)函数，清除所有在timeStamp时间之前的记录，你会怎么修改你的设计。实现。时间复杂度是多少。

```
unordered_map(key, pair<expireTime, value>)
Follow up: map(time, pair<key, value>) unordered_set(key)
template<typename K, typename T>
class M {
private:
    map<K, pair<T, int>> dict;
public:
    void put(K key, T value, int expireTime) {
        dict[key] = make_pair(value, expireTime);
    }

    T get(K key, int curr) {
        if(dict.find(key) == dict.end()) {
            return NULL;
        }
        else if(dict[key].second > curr){
            return dict[key].first;
        }
        else {
            return NULL;
        }
    }
}
```

```

bool hasKey(K key) {
    return dict.find(key) != dict.end();
}

void cleanUp(int time) {
    auto boundry = dict.lower_bound(time);
    if(boundry == dict.begin()) return;
    if(boundry->second.second == time) boundry--;
    dict.erase(dict.begin(), boundry);
}
};


```

二面：

实现一个票数统计器。String getLeader(Collection<Vote> votes, Long timeStamp) 方法。其中Vote类有两个方法(String getLeader() 和 Long getTimeStamp())表示某一票是投给谁的，什么时候投的。

你要做的method是统计timeStamp之前所有的有效投票，把票数最高的参选人名字返回。

follow ups: 1.写一个 List<String> getTopKLeaders(Collection<Vote> votes, Long timeStamp, int k) 方法，返回某个时间点前统计出来的前k名最强领导人。 2.优化新函数时间复杂度。

<http://www.1point3acres.com/bbs/thread-314016-1-1.html>

问题是，两棵树，如果其中一棵树，可以通过交换其任意数量的节点的left child 和right child，从而得到另一棵树的话，我们就称两颗树相似。题目要求你写一个判断两棵树是否相似的函数（返回true / false）。

楼主用比较intuitive的递归做的，对于每个节点，先不交换left child和right child，递归call，如果递归返回false，再交换left child和right child递归call。然后问这个算法的复杂度是什么，楼主只是觉得这个复杂度肯定高于线性，但是一口也答不上来是多少，然后凭感觉说了个n方，居然蒙对了。。在另一个面经帖也看到了这题，计算方法可以通过递归式 $T(n) = 4 T(n/2) + c$ 推导出。

面试官从头到尾都没有表现出情绪，只是说了个good，然后follow up让我优化这个算法。这题其实通过bfs，用一个map记录每个节点的parent，可以优化到 $O(n)$ ，但是当时大半夜的楼主头脑也不太清晰，隐约觉得可以用bfs，但是map这没转过来，想了很久，然后面试官提示不一定输入的是树的root，可以destroy这个树，然后楼主想到可以把输入转化成list of edges的形式（做过OA的应该明白），比如一个 ( a ( b ) (c (d) (e))) 这棵树可以转化成 [(a,b), (a,c), (c,d), (c,e)]这种形式。如果两棵树相似的话，那么它们list of edges里的元素必须完全一样。其实后来一想这个做法和bfs的原理也是相同的，其实每个edge的表示就是对应map的每一个键值对。然后面试官说了个good，然后就让我问问题了。楼主这轮当时感觉特别不好，一个是迟到了，再一个感觉题目不难，只做了一题，并且follow up还要了提示。。。

直接做题。假设让你随机生成一个迷宫，你准备怎么做。楼主先答了个naive的，比如二维数组，在里面随机地放-1作为walls，但是这样会出现问题比如没有有效路径，或有效路径太多等等。然后大叔让我保证有个有效路径，楼主就想到可以dfs随机生成一条左上角到右下角的路径，然后在这个路径的基础上，用dfs随机生成一些无效路径。但楼主突然想到这条路径也有可能向中间发展而到不了右下角，大叔随口一说用backtracking不就完了，楼主恍然大悟，对呀就backtracking。。大叔说good，然后让我忘掉迷宫，说现在给你一个长方形，让你从左上角到右下角生成一个随机路径，写一个函数，楼主心想这不就是迷宫嘛（= =||），就用dfs+backtracking写，每一点的四个方向用一个数组表示，随机shuffle一下就行。写完后大叔觉得有bug，楼主解释了一下说服了大叔，感觉大叔比较满意了，就开始让我问问题了。这轮还比较满意，结合第一轮表现以为要加面的，没想到直接送hc还过了，真的是运气不错。。。。

```

vector<pair<int, int>> mazeGeneration(vector<vector<bool>> matrix) {
    vector<pair<int, int>> path;
    dfs(matrix, 0, 0, path);
    return res;
}

void dfs(vector<vector<bool>> &matrix, int i, int j, vector<pair<int, int>> &path) {
    if(i == matrix.size() - 1 && j == matrix[0].size() - 1) {
        res = path;
        return;
    }
    matrix[i][j] = true;

    array<array<int, 2>, 4> pos;
    pos[0] = {i, j + 1};
    pos[1] = {i, j - 1};
    pos[2] = {i + 1, j};
    pos[3] = {i - 1, j};
    shuffle(pos);

    for(int k = 0; k < 4; k++) {
        int x = pos[k][0];
        int y = pos[k][1];
        if(x >= 0 && x < matrix.size() && y >= 0 && y < matrix[0].size() && matrix[x][y] == false) {
            path.push_back(make_pair(x, y));
            dfs(matrix, x, y, path);
            path.pop_back();
        }
    }
}

```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313468&page=1&authorid=258683>

第一轮：

东欧小姐姐，全程态度很好，一直很耐心带着笑容，体验很棒。

上来直接做题，第一题warm up, transpose a square matrix in place.

完了说“ok this seems too easy for you”，心想完蛋，要出难题了..

第二题check if it is possible to reconstruct an array of unique numbers of size N given a set of its subsequences.

看题就很懵逼，要了好多example才明白：

1->9->5

|->3

比如[1,9,5]和[1,3,9]可以reconstruct成[1,3,9,5], 而[1]和[2]不能construct成[1,2]因为不知道1和2谁先谁后, [1,9,5]和[3,9]也不行，因为不知道1和3谁先谁后。必须得要是construct出一个没有ambiguity的sequence才行

想了好一会儿完全没有头绪，在多重提示下才知道要用graph, 然后我想到说要find a single path that visit every node exactly once. 小姐姐表示ok，开始写码，dfs写得bug略多，自己都蒙圈了，在几个提示下改了回来，但是自己其实还是很懵逼...然后超时5分钟，我只是写出了dfs的helper，没有写怎么call，小姐姐说待会儿你还有下一个面试，i get your idea了，就问有什么问题了..

有向图欧拉回路，看所有点是不是入读等于出度（除了2个点，起点和终点）

Follow up 估计是求出具体路径，用dfs遍历每条从起点到终点的路径，找出其中的euler path.

第二轮白人Ben，上来先介绍一波自己做什么的，大概就是和search和backend有关，然后开始做题  
路口的原题三司令，直接默写了下来。然后ben说自己不熟悉Python，于是从头到尾一行行读了代码，说it looks good to me了才继续

follow up这个string会很长很长，memory里面存不下，一次只能读一个character, 也就是说把原题的string输入改成输入一个function getChar()

我一开始说应该还是相似的算法，只不过是要keep一个当前指针，然后用一个while loop写

他说问题还是没有解决，于是又给hint说你要改一下dictionary里面存的东西，一开始是存每个character的count，又是在重重提示下才说要改成存这个字母的latest index, 然后每次当dictionary里有超过k个key的时候呢，就找到value最小那个key把它删掉，然后长度就是当前idx减去最小的value。。。然后这里我贪图代码简洁，用了min(dict, key=d.get), 他又有点没看懂，于是又说了好一会儿...于是45分钟就到了

## Hashmap 和treemap 互查

```
class Solution {
public:
    int lengthOfLongestSubstringKDistinct(string s, int k) {
        if(k == 0) return 0;
        unordered_map<char, int> c2time;
        map<int, char> time2c;
        int maxLen = 0;

        int j = 0;
        for(int i = 0; i < s.size(); i++) {
            if(c2time.find(s[i]) != c2time.end()) {
                time2c.erase(c2time[s[i]]);

            } else {
                if(c2time.size() == k) {
                    auto out = time2c.begin();
                    j = out->first + 1;
                    time2c.erase(out);
                    c2time.erase(out->second);
                }
            }
            time2c[i] = s[i];
            c2time[s[i]] = i;
            maxLen = max(maxLen, i - j + 1);
        }

        return maxLen;
    }
};
```

全程感觉他的语气很冰冷，没有第一轮那种温暖的感觉，只做了一个题，然后第一轮的题也没写完，大概率是要GG了。不过其实内心挺舒服的，确实技不如人，但是感觉发挥出了正常水平，给的hint也都能一点点地往前摸索，没有完全地stuck也没有说很紧张，所以也没什么遗憾了。Google还真是这个世界上最伟大的科技公司之一，确实得要足够聪明和努力才有机会吧。楼主自觉天生愚笨，逻辑思维不强，刷题这么久以来碰到非典型题也还是转不了弯，但是能看到比刚开始时要好很多了吧，所以还是要继续努力，与各位共勉

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313855&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

面试官美国小哥，上来二话不说直接问题

( 1 ) 给你一个公司里员工的上下层关系 ( 数据结构是什么面试官要我自己决定，我用的是n叉树 )

output 公司里ceo ( root ) 与最下面的员工 ( leaf ) 的最大层数差 ( 找最长的一条路径返回长度 )

由于数据结构是树，直接dfs搞定

```
struct TreeNode {  
    vector<TreeNode*> children;  
};  
  
int deep(TreeNode *root) {  
    if(root == nullptr) return 0;  
  
    queue<TreeNode*> q;  
    q.push(root);  
    int deep = 0;  
  
    while(!q.empty()) {  
        deep++;  
        int size = q.size();  
        for(int i = 0; i < size; i++) {  
            TreeNode *front = q.front();  
            q.pop();  
            for(auto child : front->children) {  
                q.push(child);  
            }  
        }  
    }  
    return deep;  
}
```

( 2 ) follow up，现在每一个员工都被assigned了一个幸福值，写一个方法返回所有员工的幸福值

幸福值 = 员工本身的幸福值 + 所有该员工下属的幸福值的平均值 ( 这里下属既可以是直属也可以是非直属，在n叉树里表示的是什么应该都知道吧hhh )

同样用dfs，但是每次递归返回的时候要包含一些信息，比如，有多少个下属员工，下属员工的幸福值总和是多少之类的

```
struct TreeNode {  
    vector<TreeNode*> children;  
    int val;  
};  
  
class Solution {  
    unordered_map<TreeNode*, int> happy_index;  
    unordered_map<TreeNode*, int> relation(TreeNode *root) {  
        dfs(root);  
        return happy_index;  
    }  
    void dfs(TreeNode *root) {  
        if(root == nullptr) return;  
        for(auto child : root->children) {  
            relation(child);  
            happy_index[child] = root->val + relation(child);  
            relation(root);  
        }  
    }  
};
```

```

//num, sum
pair<int, int> dfs(TreeNode *root) {
    if(root == nullptr) {
        return make_pair(0, 0);
    }
    if(root->children.size() == 0) {
        return make_pair(1, root->val);
    }

    int num = 0;
    int sum = 0;

    for(auto child : root->children) {
        pair<int,int> happy = dfs(child);
        num += happy.first;
        sum += happy.second;
    }
    happy_index[root] = num / sum;
    return make_pair(num, sum);
}
};


```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313728&extra=page%3D1%26filter%3Dsorid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮：

一个白人小哥，稍微迟到了几分钟。上来直接做题。Ic 三司令。没啥印象了。先写了个brute force，然后让我优化。想了一会儿，才想起来sliding window的方法。写完后，问了下时间复杂度，然后就没时间了。。

第二轮：

不知道哪儿的小哥，有点口音。在下英语渣，基本每个问题都要他重复一下。。

题目是之前看过的面筋，给两个list，求第二个list独有的元素。先是是没有duplicates的情况，然后是有duplicates的情况。分别用set和map解决。

写完还有30分钟左右。然后开始discussion。问我hashmap的操作时间复杂度，我说是 $\log(k)$ , k是 key的长度。他不满意，想了一会儿，我说通常我们可以默认是 $O(1)$ , 他说Ok, 这是我想要的。。

然后讨论了一下hashmap 的collision并让我优化，我讲了一下常用的处理方法。小哥又问假设我们已经有了perfect hash function，再怎么优化。我说有collision的时候底层是存了链表，但我想

不到很好的能同时优化insert和query的方法。他说优化query的 $O(k)$ 就行，并暗示要 $\log n$ 的复杂度，于是我说binary search，他说可以。然后，他说没时间做下一题了，稍微聊了聊就结束了。

题目不难，但依旧每轮只做了一题。。。希望给个机会吧。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312999&extra=page%3D2%26filter%3Dsorid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

应该是一个国人大哥

只有一道题。

给x,y轴上一堆点，点的坐标都是正整数，问这些点是否是关于x = ? 的直线对称。

大哥问了半个多小时简历，所以只有一道题了。。

```
class Solution {
public:
    bool isReflected(vector<pair<int, int>>& points) {
        if(points.size() == 0) return true;

        unordered_map<int, unordered_set<int>> dict;
        int max_p = INT_MIN;
        int min_p = INT_MAX;
        int first_y = points[0].second;

        for(auto p : points) {
            dict[p.second].insert(p.first);
            if(p.second == first_y) {
                max_p = max(p.first, max_p);
                min_p = min(p.first, min_p);
            }
        }

        double line = (min_p + max_p) / double(2);
        for(auto points_in_line : dict) {
            unordered_set<int> st = points_in_line.second;
            for(auto p : st) {
                if(st.find(2 * line - p) == st.end()) {
                    return false;
                }
            }
        }
    }

    return true;
}
};
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313276&extra=page%3D2%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

给两个数n和sum，要求输出一个数组，长度为n，元素的和为sum，并且值都是随机产生的，这里对随机的定义是多次调用这个函数，输出的值不应该含有任何规律或顺序

比如说f(n, sum), n = 2, sum = 4，连续调用，那么输出可能是[0, 4]，[1, 3]，[2, 2]，[3, 1]，[4, 0]中的任何一个，每次输出一个满足要求的数组就行

问了可以使用库函数产生随机数后就写了，有点bug小哥指正了一下立马改了，感觉还行

我的办法是前n - 1个数随机，最后一个数就是sum - (n - 1)个数的和，问他这样算不算破坏了输出的随机性，小哥说不算，然后就part2了

Part 2 :

在上一问的解的基础上，给一个max参数，要求输出的数组中的值不大于max值，还用上面那个例子，f(n, sum, max), n = 2, sum = 4, max = 3, 那么上面的解中[0, 4]和[4, 0]就是不合法解了

在上一问的基础上先判断是否有合法解，然后继续用上问的代码，在最后一个数的判断上，如果大于max，随机选前面的随机个数增加确保不大于max基础上的随机的值，直到最后一个数合法为止

越写bug越多，最后小哥说：I know you will get the correct answer if you continue, but the time is below...(好像是这么说的当时还在写没怎么听)

问了问这问错了的影响，他说you got the first one and this is really a hard problem...

我读书少，感觉这题除了我的解法太暴力导致写的很蠢应该没什么算法吧，他这么说会不会是暗示我凉了？

另外，我不知道怎么算随机啊，肯定应该有更简单的解法吧

最后，如果网上有出处的话楼下发一下哈，我也想知道不蠢的做法

```
vector<int> random(int n, int sum) {
    vector<int> buckets(n);

    for(int i = 0; i < sum; i++) {
        buckets[rand() % n]++;
    }
    return buckets;
}

//1-10 2,3,4
vector<int> random2(int n, int sum, int cap) {
    vector<int> buckets(n);
    unordered_set<int> blackList;

    for(int i = 0; i < sum; i++) {
        int rnum = rand() % (n - blackList.size());
        int idx = 0;
        for(int j = 0; j < n; j++) {
            if(idx == rnum && blackList.count(j) == 0) {
                buckets[j]++;
            }
            if(blackList.count(j) == 0) idx++;
        }
        if(buckets[rnum] == cap) {
            blackList.insert(rnum);
        }
    }
    return buckets;
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312677&extra=page%3D2%26filter%3Dsorid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮是设计题，给你函数 int timerID = startTimer(int milliseconds, Runnable run); cancelTimer(int timerID); 还有一个class sprite 有draw(int x, int y) function，功能是在一个画板上让你在x y这个坐标画出这个sprite

要求让你实现一个单线程animation画板，画板大小1024x1024，要求你在(512, 512)处画一个地球，然后画一个月球，月球每五秒绕着地球转一圈，轨道半径是125。

要实现的function有

```
start(){}
stop{}
```

```
draw(){}  
面试官人真的非常好，我一个转专业的，对线程什么的一无所知，能感觉到他在努力帮助你写出来。我全程都在疯狂clarify问题  
，因此写的时间不是很多，代码也有很多毛病。。。回来仔细一想，还有一些bug。。。move on 了，说实话面试官这么帮助  
我，我感觉面试体验还是挺好的，即使跪了也觉得学到了东西。
```

## 第二轮

这一轮很凉，先出了一道很简单的is similar tree，easy 难度吧，感觉就是个warm up.

```
      a           a  
    b   c       c   b  
  d e f g     g f d e
```

这两颗是similar tree,因为把孩子flip一下就变得一模一样，geek for geek上也有原题。

然后问了时间复杂度，我懵逼了。。。。应该是n方

$T(n) = 2T[n/2] + 2T[n/2]=2^2\log n$

然后难的地方来了。。。。。。。让我设计一个md5 hash function类似的function,来判断两棵树是否是类似的，不一定要做到完全accurate....然后我说把每一层level的和加起来返回一个int, 比较int是否一样，显然不是小哥满意的答案。

狗家面试确实比较看运气，发个帖子求个好运吧。

```
size_t encodeTree(TreeNode *root) {  
    size_t root_hash = hash<int>{}(root->val);  
    if(root == nullptr) return 0;  
    if(root->left == nullptr && root->right == nullptr) return root_hash;  
  
    size_t left = encodeTree(root->left);  
    size_t right = encodeTree(root->right);  
  
    if(left > right) return (left >> 1) ^ right ^ root_hash;  
    else return (right >> 1) ^ left ^ root_hash;
```

}

我昨天也面到了tree这个题，写的也是recursion的办法，面试官问能不能优化这个算法，我想了半天，他提示可以destroy这个tree，你函数的input不一定是root，然后我就说可以把input变成list of edges的形式，比如 ( A (B)(C))可以转化成[ ( A C ),(A B)]，如果tree是相似的，这个list of edges内容应该是相同的。不知道这个list of edges排序后能不能作为楼主这题的hash

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312689&extra=page%3D2%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

一面：

扫地机器人，地里面有有不说了。

```
class Solutions {  
private:  
    int direction;  
    bool move(Robot &bot, int x) {  
        int turnRightCount = (x - direction) % 4;  
        for(int i = 0; i < turnRightCount; i++) {  
            bot.turnRight();  
        }  
    }
```

```

        return bot.move();
    }
    void dfs(Robot &bot, int x, int y, unordered_set<pair<int, int>> &visited) {
        bot.clean();
        visited.insert(make_pair(x, y));
        pair<int, int> up = make_pair(x, y + 1);
        pair<int, int> down = make_pair(x, y - 1);
        pair<int, int> left = make_pair(x - 1, y);
        pair<int, int> right = make_pair(x + 1, y);

        if(visited.find(right) == visited.end() && move(bot, 1)) {
            dfs(bot, x + 1, y, visited);
            move(bot, 3);
        }
        if(visited.find(down) == visited.end() && move(bot, 2)) {
            dfs(bot, x, y - 1, visited);
            move(bot, 4);
        }
        if(visited.find(left) == visited.end() && move(bot, 3)) {
            dfs(bot, x - 1, y, visited);
            move(bot, 1);
        }
        if(visited.find(up) == visited.end() && move(bot, 4)) {
            dfs(bot, x, y + 1, visited);
            move(bot, 2);
        }
    }
public:
    void cleanRoom(Robot &bot) {
        unordered_set<pair<int, int>> visited;

        dfs(bot, 0, 0, visited);
    }
}

```

二面：

上来问了问 如何提升网络速度， 我说的就是分布式啥的。

然后转换字符串， python的命名规则转java的。

follow up:就是把一个代码的python形式的命名转换成java的

加面：

第一个题目是给你一个array,假设叫nums. 那么要输出一个array,假设叫nums2 , nums2[i][j]就是 nums2[i][j] = nums[0..i][0...j]。

```

void matrixSum(vector<vector<int>> &matrix) {
    for(int i = 0; i < matrix.size(); i++) {
        for(int j = 0; j < matrix[0].size(); j++) {
            int sum = 0;
            if(i > 0) sum += matrix[i - 1][j];

```

```

        if(j > 0) sum += matrix[i][j - 1];
        if(i > 0 && j > 0) sum -= matrix[i - 1][j - 1];
        sum += matrix[i][j];
        matrix[i][j] = sum;
    }
}
}

```

第二个题目是给一个queue，queue中有node，node中是时间戳和字符串。其中queue中的node是按时间戳增序排列。输出一个queue，要求，对于当前的node，如果这个node的字符串在最近的10s中被加入到了output queue中，就不能加到output中，否则可以。

Time, string

我采用的就是hashMap<String, Integer>, 记录每一个string被加入output的最近时间，然后对于新来的node, 来决定加入output还是删掉

follow-up是queue特别特别大，要我优化空间，我就是用了一个TreeMap<Integer, String> 和 set<String>。这两个数据结构用来存储最近10s中被加入到output的string，然后对于新来的node，判断能不能加入到output中

```

queue<Node> search(queue<Node> q) {
    unordered_map<string, int> dict;
    queue<Node> res;

    while(!q.empty()) {
        Node top = q.front();
        q.pop();
        if(dict.find(top.content) == dict.end() || top.time - dict[top.content] > 10) {
            dict[top.content] = top.time;
            res.push(top);
        }
    }
}

return res;
}

```

设计题是假如你有一个id machine，要分配很多很多的unique id给requester machine，怎么办。

我说分区，不同区域用一个id machine管理。然后每个id machine管理的id set不相同。

然后他问如果id machine 1 有 id {1,2,3}，然后分配给了两个requester machine {0,1}，然后这个时候id machine 需要重启，重启后另一个requester machine请求id，问应该怎么办。

我说的是弄个备份，一旦id machine 坏了，从备份里面提取数据，然后再将剩下的id分配给新的请求。

我不知道为什么加面的题目这么简单，但是我感觉和面试官沟通的不好，可能自己的口语还是太差了。。。

我看网上说good luck就是没戏了，可能真的需要move on了。。。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312546&extra=page%3D2%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

今早面完Google，发面经攒一下人品。

1. 一面听口音是一个印度？南美？的小姐姐，整个面试只有四十分钟。她在机器学习领域工作，因为我简历上写了很多关于ML的project，所以针对我的项目她问了很多问题。后来剩的时间不太多了，出了一道经典的换汇率问题。然后问了几个问题就结束了。

2. 二面是一个美国小哥。上来自我介绍后开始做题。题目是这样的，假设有一条高速公路，路面上有n辆车，每辆车有不同的整数速度，但是都在1-n范围内。现在给你一个数组，代表每辆车的速度。车辆出发顺序即数组顺序，问最后可以形成几个集群，每个集群的size是多少？可以理解为，虽然车辆速度不同，但是即使后面的车比前面的车速度快，因为不能超车，最后肯定只能以前车的速度行驶，这就形成了一个集群。比如[2,4,1,3]，最后[2,4]是一个集群，[1,3]是一个集群。

Follow up是现在假设想再加入一辆车，这个车的速度比其他车都大，但是不确定这个车的出发顺序，让输出最后所有可能的每个集群的大小（List of List）。要求是可以调整并调用之前的函数，但是只能调用一次。思路还是比较直接的，但是当时太紧张了，不知道怎么去实现，耽误了好久，写到最后发现有bug...但是小哥好像没发现== 最后问了几个问题结束了。

```
vector<int> getCluster(vector<int> &speeds) {
    vector<int> res;
    int curr;
    for(int i = 0; i < speeds.size(); i++) {
        if(i == 0 || speeds[i] > speeds[i - 1]) {
            curr++;
        }
        else {
            res.push_back(curr);
            curr = 1;
        }
    }
    if(curr != 0) res.push_back(curr);
    return res;
}

vector<vector<int>> getCluster2(vector<int> &speeds) {
    vector<int> noHighSpeed = getCluster(speeds);
    vector<vector<int>> res;

    noHighSpeed.insert(noHighSpeed.begin(), 1);
    res.push_back(noHighSpeed);
    noHighSpeed.erase(noHighSpeed.begin());

    for(int i = 0; i < noHighSpeed.size(); i++) {
        noHighSpeed[i]++;
        res.push_back(noHighSpeed);
        noHighSpeed[i]--;
    }

    return res;
}
```

总体来说，题不难，考的挺灵活，最重要的还是放平心态。面试官人都很好，看到你有困难会给你hint，所以一定要记得多交流，多说自己的想法。啊啊啊啊啊啊，最后，求进pool！！

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=313437&extra=page%3D2%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

印度小哥介绍一下在学校，学什么，项目

第一题，利口已刘玲

第二题，给一个数组，长度是N，数字从1-N，问可以最多形成几个chunk，sort完每一个chunk以后可以得到sort的array

比如 [ 2 , 1 , 3 , 4 , 5 ]

最多可以形成3个chunk [ 2 , 1 ] [ 3 ] [ 4 , 5 ]

```
int sortedChunk(vector<int> &nums) {
    vector<int> right(nums.size());
    int min_right = INT_MAX;

    for(int i = nums.size() - 1; i >= 0; i--) {
        if(nums[i] < min_right) {
            right[i] = nums[i];
            min_right = nums[i];
        }
        else if(i != nums.size() - 1){
            right[i] = right[i + 1];
        }
    }

    int count = 0;
    for(int i = 0; i < nums.size(); i++) {
        if(nums[i] == right[i]) count++;
    }

    return count;
}
```

.

follow up :

假如数字不是从1-N，怎么做

第二题两问都没有做出来，在提示下才写出了代码，跪了，很难过

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315094&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%255B3046%255D%255Bvalue%255D%3D1%26searchoption%255B3046%255D%255Btype%255D%3Dradio&page=1>

给两个char array，其中有一些char为backspace就是删除前面的字符，要求输出一个boolean判断两个char array是否相同，时间要求O(n)，空间要求O(1)

例如：

[a,b,'\\b',d,c] 和[a,d,c] true

[a,b,'\\b','\\b','\\b',d,c] 和 [d,c] true

[a,b,d,'b'] 和 [a,d] false

先给了用Stack的方法，TimeO(n), SpaceO(n)，没让写code

之后要求TimeO(n), SpaceO(1)，po主试着从前往后parse没成功，好久之后小哥给提示从后往前parse试着写，po主渣，没写完

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314883&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮：2道题。第二题可以看作是第一题的extension。

第一道：找到最长的字符串的长度，要求这个子字符串的头字符和尾字符相同。

如abdcbe，则最长的是bdcb，头b和尾b相同，返回4。要求时间复杂度O(n)。

第二道：找到最长的连续子序列，要求这个连续子序列的和为0。

先用sum\_array转换一下，就变成了第一道题。

**要考虑好edge case。这个很重要！**

第二轮：1道题。OOD。

设计一个扫地机器人。

这个机器人可以前进，向左转，向右转。

方法有move, turnRight, turnLeft, clean()。

设计相应的数据结构使得它能扫完整个房间。房间大小未知，机器人的初始位置未知。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314342&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

back to back两轮电话面试，两轮都是美国小姐姐

第一轮：

给一个棋盘和一个可以走的步数，return所有路线。

input : int[] board, int number\_of\_steps [1point3acres.com/bbs](http://www.1point3acres.com/bbs)

就这一个题，第25分钟做完就让问问题了，为什么这么快呀 0.0 担心&#128531;，第一轮很紧张呃

第二轮：

LC 443,

给一个二叉树，找到坏的地方

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312170&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

问了一道题

给一个数组，n的长度（ $n \geq 1$ ），含有所有 $1 - n$ 的数字，要求得到一个最小的index，满足条件：如果把数组小于等于index的部分排序好，并且把数组大于index的部分排序好，那么整个数组就排序好了。

看题目可能有点绕，以下是当时给的两个examples：

3 1 2 5 4

返回的index为2，即如果(3,1,2)和(5,4)分别排序完成，那么整个数组就排序好了。

2 4 3 1

返回的index为3

注意由于含有所有的 $1-n$ 的数字，所以数组里是没有重复元素的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314139&extra=page%3D1%26filter%3Dsorid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮，一个很体贴的白人小哥，领我去面试地点的路上还问我来这边感觉怎样，面试中也一直跟我说别紧张不要panic，虽说我也并没有panic。。。

先问了一个题是给一个list和一个set，都是integer的，然后问list里有没有contiguous sublist和set完全match，就是数字和出现次数都一模一样的，写一个boolean function，楼主给了一个 $O(n)$ 的算法，写完之后他又让我想了几个不同类型的 testcase去测了一下代码，然后又问我还会不会有没测到的其他情况的 testcase，我就大概给他讲了下这个题的 testcase 可能会分成几类，每一个都说了下我写的代码怎样能work的，然后面试官听完夸了一下我的代码很elegant，我心里正高兴，结果他就说来我们再做一道题，我没想到只剩不到15分钟还能再问我一个新题，当时大脑就有点宕机。

第二题是给一个2D T/F matrix，然后让判断其中是否存在一个四个角都是true的矩形，确认了一下是必须要四个点单行单列的不算，然后我因为当时脑袋宕机了一开始死活想不到最优解，就愣了近一分钟，然后小哥就主动问我有什么想法了，然后我就先给了一个 $O(n^2m^2)$ 的暴力解，在讲的过程中又想到一个 $O(nm^2)$ 的优化，又说了，然后后来又想到一个可能再优化一些但是也还是到不了 $O(nm)$ 的，想着说了总比不说强吧，但那阵已经没什么时间了，下一个面试官也到了，小哥就一直在pack up他的东西，也不知道听进去了多少，不过小哥临走前还一直在祝福我希望他是没有打算挂我吧。。

```
struct hasher {
    size_t operator()(pair<int, int> p) const {
        return hash<int>()(p.first) ^ (hash<int>()(p.second) >> 1);
    }
};

bool foo(vector<vector<int>> &grid) {
    unordered_set<pair<int, int>, hasher> dict;

    for(int i = 0; i < grid.size(); i++) {
        for(int j = 0; j < grid[0].size(); j++) {
            for(int k = j + 1; k < grid[0].size(); k++) {
                if(grid[i][j] == 1 && grid[i][k] == 1) {
                    pair<int, int> p = make_pair(j, k);
                    if(dict.find(p) != dict.end()) return true;
                    dict.insert(p);
                }
            }
        }
    }
}
```

```
        }
    }
    return false;
}
```

第二轮，很幸运的碰上了一个国人大哥，一上来先问我有没有做过LC的一道hard题，他说他一般会问那个，然后很不幸楼主没做过，也照实说了，本来以为是要做的，结果大哥说那行那我给你换个medium的吧，LC拾贰，整个写完之后大哥就直接让我看他电脑抄下来的我的代码给他讲一遍，多亏是让我看他那个讲，中间我才发现他有几个地方打错了，一度怀疑我自己的字有那么难辨识么，所以这里也给大家提个醒，白板一定把每个字母符号都写清楚了，代码会被面试官抄下来的，我当时就是有个减号和等号一起的写的太近了结果大哥一开始只抄了等号，差点整个代码就不对了。然后讲完之后时间还是剩了很多，大哥说一般这种时候面试官都会再问一个，所以看来一轮两道题是常态啊，不过后来因为这个大哥说他也不想为我就没有加题，就变成聊了十几分钟的天。。

中午就是被一个台湾大叔领着去吃了饭，说实话我们去的那个cafe其实不怎么好吃，不过我本来一开始也没打算吃太多怕下午脑子不灵光，中间聊了些谷歌平常生活工作情况这些

第三轮，又一个白人大哥，上来先给了一个定义说两个string是buddies如果他们只有两个字母不一样且这两个字母交换后就可以变成一样的，比如converse和conserve这样，然后让写一个func判断两个string是不是buddies，楼主就给了一个O(n)的算法，写完之后大哥又问我我要测我的代码的话我会选哪些 testcase，感觉谷歌对这方面好像很热衷啊，要不我这就也是巧了，然后我就大概说了下都有哪些不同类型的case要考虑，然后一看时间又剩好多，估计又要被问其他题，这次在听到大哥说我写得很快时就不敢放松了，感觉这简直是谷歌的套路，然后就来了一个类似上一题的延展那样子的，就是说有n个字母不一样且交换完能相同的就叫nth buddies，然后让我写一个int func判断两个string是nth buddies，比如listen和slient这种，然后我一开始因为想看看能不能有个既省时间又省空间的算法，就又沉默了一分钟，然后这个大哥就也主动问我有没有什么想法了，感觉谷歌真的很注重交流，然后没办法我就把我最开始想到的一个O(n)的算法给说了，后来也写了代码，看大哥也没有再问我优化的样子，我也就没再想了

第四轮，一个亚洲人，不确定是不是国人，不过就算不是应该也是离得不远的那种。。给的题是输入一个BST返回min difference，这个大哥感觉很注重思路，我给了解法之后和我进行了十分之细致的探讨，每一步怎么实现用什么结构怎么遍历全部问了，然后还让我先给他写个例子，然后我一开始写了个三层的他还嫌太简单让我给一个更complex的，然后把这个例子从下到上跑过每一层返什么值了variable怎么改变了全都写过之后才让我写代码，然后写的过程中我有一次改动他还让我解释了为什么改动为什么觉得之前那个写法不好，真的是问的很细，全部写完后又让我从头梳理了一遍，啊对，然后就又是例行的问我 testcase都会选哪些，然后四十五分钟就过去了。。这真的是我面的时间最长的一道题。不过因为最后一轮后面也没事了，就说没关系我们不用管时间就让我问他问题，然后大哥真的是很能聊啊我每问一句他都能说个三四分钟那样子，然后我们就说了十五分钟。。后来就被大哥领去打车了，还确认我没问题了他才走

整体感觉下来就是谷歌的面试官人都是很nice很热心的，然后也很注重互动和交流，你要是不说话他们就会主动问你的，然后我这几个感觉怎么test代码问的很多，对corner case的处理啊这些，然后就是千万不要答上一道题就放松，我第一轮的惨痛教训啊。。再就是不必着急写代码，我反正每次都是他们先跟我把思路整个聊完觉得可以了就会跟我说你可以写代码了这样子

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314930&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

刚面完GG,399, evaluate division,就是把字母换成了币币之间的汇率,建个map 存下edge, graph dfs 或者bfs，搜索一下。gg的风格感觉变了，不能一上来就写题，他会要求我问一些关于这个input，output的问题，感觉这个答得不好。很快写好代码，面试官特意强调不写代码，接下来讨论，如果有条路径都能从起点到终点，但是得出的转换汇率不同，这种情况你怎么处

理。我反正懵逼了。。当时我就以为他要找个最短路径，他说不是。之后我说那就返回这些路径结果的平均值。他又说不是..我就反问他，这题设有问题吧，这汇率转换，虽然路径不一样，但是结果应该一样的吧。之后他说不同的交易所肯定有价差的呀，他要的是，最低的汇率，和最高的汇率.... 原来他是想搬砖套利赚钱..目测面试官是个玩币的

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=312649&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

12月15号面的，刚打电话拒了，现在手头什么都没有但还是要微笑着接着找。Anyway...面试题目：  
1拜仁中年，上来说我们这轮code不会太多，先写一个基本的，然后写了检查字符串是否回文。。利口152，然后口述判断字符串是否是pangram的思路。确认这两个定义我都知道以后，开大招：给定一个dictionary，生成最短的字符串，既是palindrome又是pangram。每个单词可以用一次然后可以用无限个单词。利口336的至尊无敌加强版，也没有写code一直在画搜索树然后分析time / space，什么bfs，dfs，A\* search轮着来，他也一直说make sense，good之类。一直就是探讨。有人知道标准思路是什么吗？

咬字不清的拜仁小哥，嘴有点歪说话含糊，说：这一轮我们考察基础一些的知识，告诉我什么是二叉树，然后什么是BST。好我们做一个小热身，你不应该花费超过五分钟，count height of binary tree。我写了一个helper来完成recursion，他问为什么不在主函数，我说可以handle wrong input format raise exception之类忽悠过去。然后正题利口109，先用nlogn做了之后小哥说优化，没注意过这道题的discussion，想了半天在提示下写出了n的解法。

看似阿三没有口音的黝黑小哥，利口159，装模作样想了下滑动窗口每次挪一个然后问time complexity说是否还有优化，最后提示下想出来高票的nlogn解法。

中东大叔？10年老googler，给定一个polygon，边都是垂直和水平的，输入是所有的点组成的1个array，从最下然后最左的点开始。再给一个任意点问点是否在polygon里面，点在边上也算在polygon里面。没在利口上见过，可能孤陋寡闻如果有见过的请指出。想到的方法是从点向上竖直做射线判断有多少个边与其有交点，然后还没考虑corner case面试官就催说时间不太多我们先写出来程序。建polygon的时候有bug少考虑了最后一个点与第一个点的连线，提示下改正，然后面试官说了一个case发现无法handle，就是射线与边重合的时候，有无穷个交点然后数交点就数不过来，想了2分钟面试官表示放弃吧太难了，还有没有什么优化，想了下说可以sort边。完事。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315162&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

给一个字符串数组，将含有相同字母的字符串归并到一个组里。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314050&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

题目是 increasing triplet subsequence。  
这个应该是地里的老题了。然而悲剧的是，我并没有见过这道题，当时给我的时候，我是硬想了一个算法。时间是O(n)但是空间也是O(n)。后来在面试官的提醒下，把空间降为了O(1)，不过也没有时间写代码了，就这么结束。

第二天hr打电话说我通过了面试，准备约onsite，而且面试官给我的是solid recommendation。一脸懵逼，你们是瞎了么？

[1point3acres.com/bbs](http://www.1point3acres.com/bbs)

说一下谷歌的面试，谷歌是对技术异常执着的公司，所以对算法要求很高。通过onsite和他们套话发现，他们是知道leetcode的，而且他们内部也有题库，专门对付leetcode，所以在面试中遇到一模一样的原题是很难，大部分情况下只会遇到类似的题

目。而根据我店面的经验，谷歌不止关注算法，他们对communication skill，debug，代码解释的能力都很重视，没有给出最优解也不一定会挂。如果个人经历中有非常强的技术研发的经验，扯皮的时候聊这个是可以加分的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314690&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

刚面完。

第一轮白人，给一个有向图，图里面有label和state。state的取值范围有多种，其中有一种是end。给你任意个图的node，问能否gurantee 走到end，即他所有的neibors都必须能走到end，只要有一个return false。我carify 有没有可能出现环，他说好问题，然后让我直接返回false。我用dfs + hashSet直接秒了，然后他让我跑两个test case。最后他说应该能work，然后剩几分钟聊天了。

第二轮老中，Majority number，第一问找1/2，我从sort讲起，讲常见的sort方法以及他们的时间空间复杂度，然后讲hash Map, bit 操作。然后假装不知道vote算法，他让我进一步优化，我说我试试水涨船高。然后demo了下，写了下算法，最后问1/3的情况。每一种算法再分析下，让我继续用vote算法。最后他说pretty good。

第三轮：比较版本的新旧。给两个String,比如“1.1.1”“3.1.1”，“3.1.1”是新版本。我跟他clarify时，说了下null 和size==0的情况，他问我这两种情况有什么不同，我给他分析了stack heap结构。然后问我应该各自返回什么，我说为null时一般视为异常，然后写了try catch结构抛出异常。但是其中一个String size == 0时返回另一个String即可。他说很好。然后写代码。挨个比较，最后他问我“1.1”和“1.1.0”和“1.1.0000”该怎么办，我说那我们需要pre process或者Pro precess，让我分析哪种好，我说时间复杂度和空间复杂度都一样，所以两种方法没区别，他说对。然后我写了preprocess，把最后的0全都删除掉。

第四轮：一办BQ，问我最近的项目，最难的地方在哪里，我答最难的地方也是我最骄傲的地方是当我们的用户增加时，我的架构需要做相应的优化。然后见我使用了cache，问使用cache常见的风险是什么，我说风险是cache的数据会掉电消失，我们使用log的方式从掉电中恢复cache数据。然后剩一半时间问我假如一辆车只有一排位置，每次乘客上车后都希望能坐在最长的interver 的中间。位置是double，一提到double我就说我们要小心double的比较，然后他问为什么，然后跟他扯double的精度问题。然后分析corne case,分析null 和 empty两种情况，接着他问如果是empty的话，返回什么比较好，我说一般情况下可以返回特定值，然后他又问这么做有什么潜在的问题，我说也许这个特定值本身是有效的。然后他又问既然如此有什么更好的解决方法么，我说那我把double改成Double，返回NULL，然后他说much better.最后用mapHeap把这题秒了，然后我自己分析时间空间复杂度，他挺满意，最后让我跑两个test case。

第五轮：interval问题。这一轮最蛋疼，他首先介绍了下谷歌的分布式系统有很多的task要运行，告诉我每个task的起始时间和终止时间，要我们判断有哪些任务可以第一个执行。花了将近二十分钟的时间才把问题抽象成算法问题，然后暴力解，先sort，又跟他扯常见的sort算法及时间空间复杂度。然后两层for loop比较。最后分析时间复杂度O(n^2)空间复杂度O(1)。然后让我优化比较次数，时间就剩一点点了，没想出来，时间到了后他自己跟我说了下他的解法，我说原来就是用两个heap，他说是的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314882&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

两轮面试，每轮一题，题目是真的不难，楼主水平太差，估计是挂了。。。

第一轮题目：

给两个string，只含有'a-z'和'B'，'B'表示退格键。要求判断两个string是否相同  
follow up是只能用O(1) space

第二轮题目：

判断一个多叉树是否是unival tree，unival tree的定义是所有node的值都相同。  
follow up是输出所有是unival的subtree的个数

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315136&extra=page%3D1%26filter%3Dsorid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

1 native speaker 题目是candy crush，给一个8\*8的空棋盘，要求用0，1，2填充满，要满足横竖每相邻3个不能相同，然后返回填好的棋盘。大概思路是一行一行遍历棋盘，然后对于每一个格子检查其上面2格是否相同，如果相同则不能填上面的数字，然后检查左边两格是否相同，如果相同不能填左面的数字，然后在剩下可填的数字中选择一个填入。follow up1是输出所有可能的填充好的棋盘，follow up2是如果要求对角方向也不能有3个相同的相邻，那么需要用至少多少个不同数字填充才能满足条件，答案是5

```
vector<vector<int>> foo(vector<vector<int>> grid) {
    array<bool, 3> color = {true, true, true};
    for(int i = 0; i < grid.size(); i++) {
        for(int j = 0; j < grid[0].size(); j++) {
            if(j >= 2 && grid[i][j - 1] == grid[i][j - 2]) {
                color[grid[i][j - 1]] = false;
            }
            if(i >= 2 && grid[i - 1][j] == grid[i - 2][j]) {
                color[grid[i - 1][j]] = false;
            }
            for(int k = 0; k < color.size(); k++) {
                if(color[k] == true) {
                    grid[i][j] = color[k];
                    break;
                }
            }
            color = {true, true, true};
        }
    }

    return grid;
}

//for follow up
void helper(vector<vector<int>> &grid, vector<vector<vector<int>>> &res, int i, int j) {
    if(i == grid[0].size() - 1 && j == grid.size() - 1) {
        res.push_back(grid);
    }

    array<bool, 3> color = {true, true, true};

    if(j >= 2 && grid[i][j - 1] == grid[i][j - 2]) {
```

```

        color[grid[i][j - 1]] = false;
    }
    if(i >= 2 && grid[i - 1][j] == grid[i - 2][j]) {
        color[grid[i - 1][j]] = false;
    }
    for(int k = 0; k < color.size(); k++) {
        if(color[k] == true) {
            grid[i][j] = color[k];
            if(j == grid[0].size() - 1) helper(grid, res, i + 1, 0);
            else helper(grid, res, i, j + 1);
            grid[i][j] = 0;
        }
    }
}

```

2 印度小姐姐，啥也听不懂，两道题，第一题是树的题，给一棵二叉树，每个节点包含自己的 value 和此节点的所有子树+自己一共包含多少节点，让返回 in-order 遍历顺序中第 k 个节点的 value。

eg. 4 (5)

```

3 (3)      7 (1)
1 (1) 2 (1)

```

第4个节点的值就是2。思路是利用每个节点存的子树节点个数，每次和k对比决定选择左子树继续遍历还是右子树继续遍历。

```

int helper(TreeNode *root, int start, int k) {
    int numleft = root->left == nullptr ? 0 : root->left->num;
    if(start + numleft == k) return root->val;
    else if(start + numleft >= k) {
        return helper(root->left, start, k);
    }
    else {
        return helper(root->right, start + numleft + 1, k);
    }
}

```

第二题是图的题0-n个房间可以放别的房间的钥匙，或者放宝藏，或者为空，求从某个房间i出发，能不能找到宝藏，可能有环。follow up 是从多个房间出发。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314867&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

昨天刚刚面试的。题目简单，不过我本人太水了，感觉要挂的节奏，因为每位面试官只问了一道题，就结束了。。。面试情况给大家 share一下吧。一面：国人哥哥，直接中文面试，倍感亲切，人挺好。莉蔻 三菱吧 没有要求更新矩阵值，但是要求数次读取，所以需要先缓存数据。之前没做此题，国人哥哥多次提示，总算讲思路讲出来了。基本功不扎实，没什么可以抱怨的。

二面，应该是印度人吧，听声音。是一个 operation parsing 的题目，类似莉蔻 二十 和莉蔻 二八二 结合的题目（或者是原题，不好意思，我刷的题不够多），反正就是一堆数字和“+”，‘-’，‘=’结合到一块，然后让 parse。思路是放进 stack，在 pop 出来，然后运算。做的很辛苦，提示多次后，才总算给出一个思路。最后一句 good luck，心凉了半截。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314522&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

一轮 1，热身题，图类，给很多状态点，有circle，给出初始状态以及目标状态，求是否有合法路径。

2，蠡叩刘吧务

而轮 依然渊踢，提耗忘记了有知道的小伙伴可以留下言 方便其他小伙伴看。给一堆乱序的数字有重复。连续的五个数字称作一个（特殊名词）。求有多少个这种（特殊名词）

伞轮 霸寇洱鎏揪

斯轮 笠鈎彑灸懿

傘巫鑾

五轮 山妖鸟

## 第一轮

只做了一道题，给一个max heap，用array表示，return 最前k大的数，但是不能modify这个heap。刚开始我说，复制一些，再一个一个取。小哥说可以，那不准你复制怎么办。我在他的提示下，想出来了。用一个size为K的新的heap，每次把当前最大的element拿出来，再把这个element的children放进去。用一个k的for loop。写完，提示下发现index有错，在提示下改对了。说了下complexity，时间到了

## 第二轮

挂就挂在这儿...

System Design，Publisher-Subscriber系统，应该是

[https://en.wikipedia.org/wiki/Publish%20%93subscribe\\_pattern](https://en.wikipedia.org/wiki/Publish%20%93subscribe_pattern)

并且有multithreading，要考虑推送的时候user 瞬间subscribe或者unsubscribe，实在是不会，写了几个class，用了extension，加了lock。真的不会...

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315015&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

首先介绍了一下做过的项目，大概10min时间吧，主要考察个人表达能力吧

然后做题，在Google Doc上写代码，问题大概是，美国大选，有很多选票，Vote，每个Vote有两个属性，一个name一个timestamp，表示投票给谁以及投票的timestamp

大概问了三个问题

1.给定一个timestamp,找到这个timestamp之前的得票最高的人，借助于HashMap实现即可，说了下思路，不需要写代码  
[1point3acres.com/bbs](http://www.1point3acres.com/bbs)

2.如果是给定一个timestamp,找到Top K的投票最多的人怎么办呢，corner case是如果有两个人投票一样怎么办(*post process, 把原投票统计里所有票数等于heap里最小票数的都加进去*)，基本思路是HashMap统计每个人得票次数，刚开始考虑用Heap来实现，找TopK，但是追问，如果投票相同的人算一名，找Top K名改怎么找，思路是用TreeMap来实现，用数量作为Key,值是List<String>，然后又追问，如果不考虑数量相同的人，怎么找Top K，除了用堆之外，还有没有其它更好的方法，答用类似Quick Sort的找第K大算法，感觉还比较满意，写代码实现一下

3.如果给定的是一个List of name , name按照投票数量从大到小排序，找到一个timestamp，使得到达这个timestamp之后，Top K的人选和给定List一致，基本思路就是先排序，然后不断的用HashMap和TreeMap做统计，如何添加和删除元素，分析时间复杂度

先将投票加入treemap ( time为key ) 然后遍历treemap不断更新hashmap做当前时间的统计，计算当前topk，比较结果

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314874&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

刚面完加面。。感觉不太好，是个印度大哥，全程处于他讲话我懵逼，我讨论他懵逼的状况，很想让他引导一下我，因为前两面的面试在我讲完思路以后都会和我讨论有什么问题，或者说细节要注意什么，能学到很多东西。。。这次好像帮助并不大TAT。。。本来准备了很多graph的题觉得算是很高频了，好不容易写6了然后一次都没被问到。

一二面的题目分别是：

1) 给一个polygon，这个polygon是以由顶点编号组成的数组给出来的，每个polygon都可以由很多三角形组成，返回组成所有三角形的顶点编号（数组）

2) 给一个可以自己生成自己的string，有规律的，输入一个数代表string里的t的个数，求生成的string，大家大概get一下这个意思

给的string是这样的: 'T' 是第一，第四，第十一等等，

这个第一第四第十一三个单词呢，分别代表的就是t这个字母在string里面的位置（这几个单词都含有t），所以这个string可以是无穷大的。面试官给了api可以直接得到数字对应的单词。

加面：

打印出所有可以被两个cube组成的不同length组成的数。cube代表的意思就是 $i^{**}3$ 这样。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314354&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

可能因为LZ是妹子吧，感觉题目真的很简单。

第一轮

给一个直角坐标系，系上给若干点，求任意两点，满足：这两点的连线将剩余点 on one side of this line, 就是说这条线可以把剩余点分到线的一边，而不存在这条线的两边都有点的情况。

一看题目很懵，但是面试官一直给hint. 先是问，给N个点可以画多少条直线？秒答  $N*(N-1)/2$ ;

对了， struct P{double x;double y;};

然后给了一个API，说 int OnWhichSide(vector<Point> line, Point p); return -1表示在一侧，return 1 表示在另一侧。

题目的function signature是( 这是我自己写的) vector<Point> PointsOnOneSide(vector<Point> input)

于是我就答把所有点遍历一遍，组成line，再判断其他点是否满足条件，如果满足，直接return。

问了复杂度  $O(n^3)$  if int OnWhichSide(vector<Point> line, Point p) has  $O(1)$ . 这题就这样结束了，全程面试官很温柔，一直good good，有bug也是让我go through一遍function自己发现的。

第二轮：

第二轮有两道题，感觉面试官没有上一个Nice，我心态就有点崩，仔细看下来觉得挺简单的，但是全程答得颤颤巍巍。

说有一个 struct Message{string source\_file;string content;}; 然后给 vector<Message> v =  
{["a","loop1"], ["b","loop1"], ["c","loop1"], ["a","loop2"], ["b","loop2"], ["c","loop2"], ["a","loop3"], ["b","loop3"], ["c","loop3"], ["a","loop4"]} 类似的vector。说要return一个新的 unordered\_map<string, vector<Message>> such that the map is like  
{{"a",["a","loop1"], ["a","loop2"], ["a","loop3"]}, {"b",["b","loop1"], ["b","loop2"], ["b","loop3"]}, {"c",["c","loop1"], ["c","loop2"], ["c","loop3"]}};

第二题是vector<Message> truncate(vector<Message> v,int num);

就是说num是多少，这个return的vector最多含有num个相同的source file. 比如对于上面那个例子，如果num是2，那么return的就是 [{"a","loop1"}, {"b","loop1"}, {"c","loop1"}, {"a","loop2"}, {"b","loop2"}, {"c","loop2"}] (最多2个a, 2个b, 2个c)。

最后发点牢骚，从12月13号开始，收到google 和 facebook 的interview，觉得自己google无望，转头去准备fb的面经和tag题，刷了2遍吧at least，最后被fb当天就拒了，google完全抱着不可能，很放松去面的，结果给了加面（说明有戏？加面到底是什么情况。。。），有点郁闷，等心情好一点再发fb面经吧

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314965&extra=page%3D1%26filter%3Dsoid%26sortid%3D311%26searchoption%5B3046%5>

蠢口恶气就

楼主当时想dfs，bfs都可以做，想都没想就直接bfs了，后来想了想好像dfs简单很多。面试官大哥估计没想到有人拿bfs解，我拿bfs写完之后一直说very good，估计心里想的是“还真有二货用bfs解这个题”。因为楼主写的c++，后来又问了问C++里string和const string区别，什么lvalue, rvalue的，具体的说不清楚，说了个大概其，大哥说还可以。应该可能大概也许能过吧。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314907&extra=page%3D1%26filter%3Dsoid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

Lucky that I got two normal questions:

Round 1:

Giving start string and end string, determine if start string can finally reach to the same as end string with below rules.

For example: [1point3acres.com](http://www.1point3acres.com)

"R L \_ \_ L R L"

"\_": the space is empty

"L": this can only swap with the empty letter \_ on its **left** side

"R": this can only swap with the empty letter \_ on its **right** side

So, "R L \_ \_ L R L" can change to "R L \_ L \_ R L", and can continue change to (if you want) "R L L \_ \_ R L"

The question is given these rules and the start string and end string, could we change the start string to end string (unlimited # moves as long as it is valid).

For example:

"R \_ \_ L R \_ R \_ L" [1point3acres.com](http://www.1point3acres.com)

can be changed to

"\_ R L \_ \_ R R L \_"

Round 2:

Q1: Just talk no coding

Given a string, how to find mode char in it?

What are some test cases?

Didn't know the purpose of this question. I asked if he looks for some sounding/efficient solutions other than counting with array/hashmap. He said good enough. lol

Q2:

Given some server to sever propagation time, and we always start with Primary server, find the longest time for propagation.

Assume no cycle, no multiple propagation paths for the same destination.

For example,

Primary server --> server A (100ms)

server A --> server B (50ms)

Primary server --> server C ( 200ms)

Primary server --> server D (10ms) [1point3acres.com](http://www.1point3acres.com)

server D --> server E (50ms)

server E --> server F (150ms)

your function should return 210ms because Primary -> D -> E -> F takes 210ms which is the longest time among all propagations.

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314851&extra=page%3D1%26filter%3Dsoid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searc>

binary tree serialize and deserialize

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314636&extra=page%3D1%26filter%3Dsoid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

刚刚结束的狗家12面。回馈地里。

一面：印度小姐姐。

只做了一道题：给一个字典和一个关键字，要求根据这个关键字把字典里可能的单词全部print出来。类似于输入法里输入几个字母然后弹出推荐的可能你需要的单词。

比如 word: "ca", dict ->{"cat", "car", "boy", "course"}, output {"cat", "car"}。

```
struct TrieNode {
    bool isWord;
    TrieNode* children[26];

    TrieNode() {
        for(int i = 0; i < 26; i++) {
            children[i] = nullptr;
        }
        isWord = false;
    }
};

class Solution {
```

```

private:
    TrieNode *root;
public:
    void addWord(const string &word) {
        TrieNode *curr = root;

        for(auto c : word) {
            int i = c - 'a';
            if(curr->children[i] == nullptr) {
                curr->children[i] = new TrieNode();
            }
            curr = curr->children[i];
        }
        curr->isWord = true;
    }

    TrieNode* getPrefix(const string &prefix) {
        TrieNode *curr = root;

        for(auto c : prefix) {
            int i = c - 'a';
            if(curr->children[i] == nullptr) {
                return nullptr;
            }
            curr = curr->children[i];
        }
        return curr;
    }

    vector<string> solver(vector<string> &dictionary, string prefix) {
        root = new TrieNode();
        vector<string> res;
        for(auto word : dictionary) {
            addWord(word);
        }

        TrieNode *withPrefix = getPrefix(prefix);
        dfs(withPrefix, res, prefix);
        return res;
    }

    void dfs(TrieNode *root, vector<string> &res, string &path) {
        if(root == nullptr) return;
        if(root->isWord == true) {
            res.push_back(path);
            return;
        }

        for(int i = 0; i < 26; i++) {
            if(root->children[i] != nullptr) {

```

```

        path += char(i + 'a');
        dfs(root->children[i], res, path);
        path.pop_back();
    }
}
};

};


```

二面：国人小哥。人非常友善。

也是一道题：有一个房间，然后有很多的event，event里有两个变量time，duration。要求这个房间里不能同时有三个event存在。

E<time, duration>

比如 E<1,3>, E<2, 3>, E<3,3> -> return false.

E<1,3>,E<2,3>,E<7,1> -> return true.

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314429&extra=page%3D2%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

题目是 boggle game, 之前没听过，就跟我大概解释了一下.

<https://en.wikipedia.org/wiki/Boggle>

题目版本比wiki上简单，就是有一个4\*4 grid, 每一个里面随机放了一个letter, 要求找出里面所有的word.

需要自己想如何存放dictionary, 第一念头用个hash table, 经提示用prefix tree (trie)效率更好.

算法就是用DFS 从每个grid 开始搜索，找到word就放到res中. 其中可以合理的使用trie, 比如每次 tmp\_word不在prefix中，就可以停止搜了. 下一个单词也不用从新搜，可以之间在trie中上一个level往下找.

```

struct TrieNode {
    bool isWord;
    unordered_map<char, TrieNode*> children;
    TrieNode() {
        this->isWord = false;
    }
};

};


```

```

class Solutions {
private:
    TrieNode *root;
    vector<string> res;
public:
    void buildTree(vector<string> &words) {
        root = new TrieNode();
        for(auto word : words) {


```

```

        addWord(word);
    }
}

void addWord(const string &word) {
    TrieNode *curr = root;
    for(int i = 0; i < word.size(); i++) {
        if(curr->children.find(word[i]) == curr->children.end()) {
            curr->children[word[i]] = new TrieNode();
        }
        curr = curr->children[word[i]];
    }
    curr->isWord = true;
}

vector<string> foo(vector<string> words, vector<vector<char>> matrix) {
    buildTree(words);

    for(int i = 0; i < matrix.size(); i++) {
        for(int j = 0; j < matrix[0].size(); j++) {
            string path = "";
            helper(root, path, matrix, i, j);
        }
    }

    return res;
}

void helper(TrieNode *root, string &path, vector<vector<char>> &matrix, int i, int j) {
    char c = matrix[i][j];
    if(root == nullptr) return;
    if(root->children.find(c) == root->children.end()) {
        return;
    }
    if(root->children[c]->isWord) {
        res.push_back(path);
    }

    matrix[i][j] = '#';
    array<int, 4> x = {i, i, i + 1, i - 1};
    array<int, 4> y = {j - 1, j + 1, j, j};

    for(int i = 0; i < 4; i++) {
        if(x[i] >= 0 && x[i] < 4 && y[i] >= 0 && y[i] < 4) {
            path.push_back(matrix[x[i]][y[i]]);
            helper(root->children[matrix[x[i]][y[i]]], path, matrix, x[i], y[i]);
            path.pop_back();
        }
    }
}

```

```
    }
}
};
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314629&extra=page%3D2%26filter%3Dsorid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮，听口音应该是个三哥，做Google map的，一开始先问我有没有问题问他，我问了两个，回答完后开始做题。

第一题，判断两个string是否是metatheses，就是其中一个string能否通过交换一对不相同的字母而得到另外一个string，比较简单，我先讲了思路，然后边写代码边描述每一步实现的作用。写完以后，面试官让我想了几个test来验证我的代码。我自己想的case都能过，他也没有提出异议，说我们先看下一道，等会儿如果有时间在来进一步讨论这个问题。

第二题，给定一个query word和一个dict（里面有一些words），输出dict里面的那个word可以又query word删除最少的letter得到，比如 dict = {"apple", "bat", "boat", "cap"}，则输入apple，输出apple，输入bonato，输出boat。面试官描述完题目后专门说这个不是edit distance。这个题也不难，可以判断query word删除多少个letter可以得到dict里面的每一个word，然后输出那个删除letter个数最少的word。我觉得利口上有原题，但记不清了，geeksforgeeks上有原题，各位有兴趣可以搜一下。我给他说了这个方法，但他问我有没有方法可以不遍历dict，我说可以建立一个dict的hashset，然后对query word进行删除letter，看得到的word在不在set里面，先删除一个letter，然后两个，三个，直到找到目标词，类似BFS。他叫我写这个不遍历的方法，我没写完时间就到了。

第二轮，也是个小哥，是做VR的，上来直接做题。

第一题，给一个BST和一个target，求出BST里比target小的最大的一个数。Geeksforgeeks有原题，其实很简单一道题，我在处理右子树的时候一个判断条件没有考虑，现在看来还是基础不扎实，面试官比较冷漠，自己就很紧张。弄了大概半个小时没完全弄出来，就是一个小问题。后来他直接说我们来看下一道题吧

第二题，这是一个系统设计题，就是出一些词和每个词的频率，设计一个方法根据频率随机返回一个词，要求速度最快。这题不难，用array根据频率存入不同个数的word，然后随机产生一个index，返回array里的word。这道题回答的还行

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314645&extra=page%3D2%26filter%3Dsorid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

1月10号的两轮电面，

第一轮是白人小姐姐很和善，只做了一道题，两个follow up

首先判断一个数字是不是lucky的(数字里不能带4，因为在中文是死的意思)，然后第一个followup是给出一个整数之前的lucky数字的数量，第二个是如何显示出一个每个unlucky的数字

100 4 14 , 24 , 34 , 40 , 41 , 42 , 43

C(9, 1)C(9,1) c(8, 1)c(9,1)c(9,1)

81 4 , 14 , ...94 40-49 44

第二轮是印度姐姐直接开始做题，一共做了三道

第一道是输入两个数组，输出他们的交集

第二道是输入两个字符串，如果他们之间的不同字符数量为1，返回true(a=xyz, b=xyu或者a=xyz, b=xyzu都会返回true )

第三道是输入不同颜色的油漆及其花费，以及一定数量的房子，要求在每种颜色油漆都会用到的前提下，以最低花费粉刷所有房子，相邻房子颜色不能一样

```
int minCostII(vector<vector<int>>& costs) {
    if(costs.size() == 0 || costs[0].size() == 0) return 0;
    int n = costs.size(), k = costs[0].size();
    vector<vector<int>> dp(2, vector<int>(k, 0));
    dp[0] = costs[0];
    for(int i = 1; i < n; i++) {
        vector<int> dp_left(k, 0), dp_right(k, 0);
        dp_left[0] = numeric_limits<int>::max();
        dp_right[k - 1] = numeric_limits<int>::max();

        for(int j = 1; j < k; j++) {
            dp_left[j] = min(dp[(i - 1) % 2][j - 1], dp_left[j - 1]);
        }

        for(int j = k - 2; j >= 0; j--) {
            dp_right[j] = min(dp[(i - 1) % 2][j + 1], dp_right[j + 1]);
        }

        for(int j = 0; j < k; j++) {
            dp[i % 2][j] = min(dp_left[j], dp_right[j]) + costs[i][j];
        }
    }
    vector<int> last = dp[(n - 1) % 2];
    return *min_element(last.begin(), last.end());
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315445&extra=page%3D1%26filter%3Dsorid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮 国人小哥 人特别好

第一题 利特口 儿把就 followup 如果数据量特别大怎么办 如果分布在多个machine上 每个机器运算性能不一样怎么办  
切，记得上下多留一行只用来计算，按性能切

第二题 给了一个数组求最小值的code 问其中某一行运行的期望运行次数 用的dp

第二轮 白人小哥

给一个grid 其中有些格子被mark了

求整个区域里所有的正方形个数 ( 正方形不能覆盖被mark的格子 )

这轮小哥迟了20分钟 + 一开始碰上题 想了好久 这轮一直在讲思路 最后得到的是时间空间都  $O(n^2)$  ( 虽然小哥说 因为他来迟了 所以就没写code 不是我的锅 但还是很虚。。。 )

第三轮 应该是国人小哥。。。

利特口 伞司令 我一开始写的nlogk的 小哥说还有 $O(n)$ 的 但是我们先看followup吧

followup 如果是个stream怎么改 写两个function 一个accept ( char c ) 一个getLength

因为我原来的代码正好可以直接用在followup上 但是accept是logk的 于是小哥要求 $O(1)$ 的 然后写了个 double linkedlist  
然后原题也顺便正好优化到 $O(n)$ 了。。。

map<int, char> unordered\_map<char, int> 记录最后一个的位置  
aabca

```
class Solution {
public:
    int lengthOfLongestSubstringKDistinct(string s, int k) {
        if(s.size() <= k) return s.size();
        if(k == 0) return 0;

        unordered_map<char, int> dict;
        deque<char> d;
        int max_len = 0;

        for(int i = 0; i < s.size(); i++) {
            if(dict.find(s[i]) != dict.end() || dict.size() < k) {
                dict[s[i]]++;
                d.push_back(s[i]);
            } else {
                max_len = max(max_len, int(d.size()));
                while(dict.size() == k) {
                    dict[d.front()]--;
                    if(dict[d.front()] == 0) {
                        dict.erase(d.front());
                    }
                    d.pop_front();
                }
                d.push_back(s[i]);
                dict[s[i]]++;
            }
        }
        return max(max_len, int(d.size()));
    }
};
```

第四轮 应该还是国人小哥。。。

给两个string 判断是否是合法的encode的

followup 如果给一个test string 和一串备选的string 判断是否存在一对合法的pair

把所有的string转成一个pattern ( 比如 apple -> abbcd banana -> abcabc ) 然后string array建trie tree 搜索的时间复杂度就是 $O(k)$ 了

( 但我隐隐感觉 小哥原本准备的答案不是这样的。。。虽然他还是让我按trie tree写了。。。然后时间空间复杂度也能接受的样子 )

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315403&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

发个今天下午Google的实习面试。感觉 Google 的面试很看人品，身边有的人面的题很难，有的面的很简单（比如我），然并卵。

第一轮：一个印度姐姐，感觉人还不错。电话刚打来就说有的事要一会再打来。接着再打来之后，出了题，然后又说有事，要挂断一会，正好给我一段时间好好思考。

题目很简单，两个List，里面的element可以是任意type。返回两个list的差。

比如 A=【1,2,2,2】

B=【1,2,3,3,3】

A-B=【2,2】

B-A =【3,3,3】

要求就是求出 A-B 和 B-A。

一开始提了一个 nested for loop 暴力解  $O(mn)$ 。她说可以。接着我提了一个  $O(n)$ ，用HashMap储存的办法，一开始我还傻傻的说要用两个hashmap，然后她提醒我一个就够了。然后我明白了就开始写代码了。接着写完之后她提出要我用  $O(n \log n)$  的时间做但是不能用hashmap，然后我就说吧暴力解里面的一个loop改成binary search就行了，她说可以。因为时间快到了就没让我写代码让我问问题。感觉她就准备了一道题。

```
pair<vector<int>, vector<int>> foo(vector<int> l1, vector<int> l2) {
    unordered_map<int, int> dict1;
    vector<int> onlyInOne;
    vector<int> onlyInTwo;

    for(auto num : l1) {
        dict1[num]++;
    }

    for(auto num : l2) {
        if(dict1.count(1) == 0 || dict1[num] == 0) {
            onlyInTwo.push_back(num);
        }
        else {
            dict1[num]--;
        }
    }

    for(auto pair : dict1) {
        if(pair.second != 0) {
            onlyInOne.push_back(pair.first);
        }
    }
    return make_pair(onlyInOne, onlyInTwo);
}
```

第二面的不好。题目也很简单，可惜我题刷的太少，不熟。面试官的英语讲的模糊不清的，很难交流。

给出一个单词的String，判断是不是decremental word，比如 Print，每次删掉一个character，然后有个help function check是否是valid word(这个不需要我写)，然后一直删到剩一个character，可以删任意character。

要求每次都得是 valid word。最后决定最原先的input的单词是否是 decremental word。  
比如 Print->Pint ->Pit ->I 这里面 每个词都是一个valid word，所以print就是一个 decremental word。

```
bool foo(string word) {
    unordered_set<string> dict;
    return dfs(word, dict);
}

bool dfs(string word, unordered_set<string> &dict) {
    if(dict.find(word) != dict.end()) {
        return false;
    }
    if(word.size() == 1 && valid(word)) {
        return true;
    }

    for(int i = 0; i < word.size(); i++) {
        string substr = word.substr(0, i + 1) + word.substr(i + 1);
        if(dfs(substr, dict)) {
            return true;
        }
    }
    return false;
}

//bfs solution
bool bfs(string word) {
    queue<string> q;
    q.push(word);

    while(!q.empty()) {
        string top = q.front();
        q.pop();
        if(!valid(top)) continue;
        if(top.size() == 1) return true;

        for(int i = 0; i < top.size(); i++) {
            string substr = word.substr(0, i + 1) + word.substr(i + 1);
            q.push(substr);
        }
    }

    return false;
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315446&extra=page%3D1%26filter%3Dsorid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

全职software engineer 电面 也不知道哪里的人 不是印度人不是中国人 不像是美国人 口音有一点 聊了十五分钟 做题，45分钟到点走人，题目如下：

给一个string 比如"abslsaasssa" 如何调整顺序 输出的是没有间隔相同的string, 比如"sasasasasbl" 如何做？

一开始想用two pointer swap，结果做了一会发现corner case太多了 于是决定greedy，建立dict，然后开始先放最多频率的char，紧接着放second highest 的char,彼此隔开 直到全部放完

做完了觉得还可以 跟内推聊了一下觉得还可以 但是晚上突然想到了corner case，应该用ceil(len(str)/len(most frequent letter))然后一位一位的放 最后把空格去掉输出str，

一周之后今天收到HR电话 拒了 冷冻期一年 没什么好说的 和apple onsite rejection一天收到的 继续找

```
string foo(const string &s) {
    unordered_map<char, int> dict;
    string res = "";
    char max_char = '*'; //check later
    int max_count = 0;
    int gap = ceil(double(s.size()) / double(max_count));
    vector<char> res_builder(gap * max_count, '_');
    vector<int> pointer(max_count);

    for(auto c : s) {
        dict[c]++;
    }

    for(auto c_pair : dict) {
        if(c_pair.second > max_count) {
            max_count = c_pair.second;
            max_char = c_pair.first;
        }
    }

    for(int i = 0; i < max_count; i++) {
        pointer[i] = i * gap;
        res_builder[pointer[i]] = max_char;
        pointer[i]++;
    }

    int i = 0;
    for(auto c_pair: dict) {
        if(c_pair.first == max_char) continue;
        int count = c_pair.second;
        while(count > 0) {
            res_builder[pointer[i]] = c_pair.first;
            pointer[i]++;
            i = (i + 1) % max_count;
            count--;
        }
    }

    for(char c : res_builder) {
        if(c != '_') {
            res += c;
        }
    }
}
```

```

    }
}

return res;
}

```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=314690&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

面试完。

第一轮白人，给一个有向图，图里面有label和state。state的取值范围有多种，其中有一种是end。给你任意个图的node，问能否gurantee 走到end，即他所有的neibors都必须能走到end，只要有一个return false。我carify 有没有可能出现环，他说好问题，然后让我直接返回false。我用dfs + hashSet直接秒了，然后他让我跑两个test case。最后他说应该能work，然后剩几分钟聊天了。

第二轮老中，Majority number，第一问找1/2，我从sort讲起，讲常见的sort方法以及他们的时间空间复杂度，然后讲hash Map, bit 操作。然后假装不知道vote算法，他让我进一步优化，我说我试试水涨船高。然后demo了下，写了下算法，最后问1/3的情况。每一种算法再分析下，让我继续用vote算法。最后他说pretty good。

第三轮：比较版本的新旧。给两个String,比如“1.1.1”“3.1.1”，“3.1.1”是新版本。我跟他clarify时，说了下null 和size==0的情况，他问我这两种情况有什么不同，我给他分析了stack heap结构。然后问我应该各自返回什么，我说为null时一般视为异常，然后写了try catch结构抛出异常。但是其中一个String size == 0时返回另一个String即可。他说很好。然后写代码。挨个比较，最后他问我“1.1”和“1.1.0”和“1.1.0000”该怎么办，我说那我们需要pre process或者Pro precess，让我分析哪种好，我说时间复杂度和空间复杂度都一样，所以两种方法没区别，他说对。然后我写了preprocess，把最后的0全都删除掉。

第四轮：一办BQ，问我最近的项目，最难的地方在哪里，我答最难的地方也是我最骄傲的地方是当我们的用户增加时，我的架构需要做相应的优化。然后见我使用了cache，问使用cache常见的风险是什么，我说风险是cache的数据会掉电消失，我们使用log的方式从掉电中恢复cache数据。然后剩一半时间问我假如一辆车只有一排位置，每次乘客上车后都希望能坐在最长的interver 的中间。位置是double，一提到double我就说我们要小心double的比较，然后他问为什么，然后跟他扯double的精度问题。然后分析corne case,分析null 和 empty两种情况，接着他问如果是empty的话，返回什么比较好，我说一般情况下可以返回特定值，然后他又问这么做有什么潜在的问题，我说也许这个特定值本身是有效的。然后他又问既然如此有什么更好的解决方法么，我说那我把double改成Double，返回NULL，然后他说much better.最后用mapHeap把这题秒了，然后我自己分析时间空间复杂度，他挺满意，最后让我跑两个test case。

第五轮：interval问题。这一轮最蛋疼，他首先介绍了下谷歌的分布式系统有很多的task要运行，告诉我每个task的起始时间和终止时间，要我们判断有哪些任务可以第一个执行。花了将近二十分钟的时间才把问题抽象成算法问题，然后暴力解，先sort，又跟他扯常见的sort算法及时间空间复杂度。然后两层for loop比较。最后分析时间复杂度O(n^2)空间复杂度O(1)。然后让我优化比较次数，时间就剩一点点了，没想出来，时间到了后他自己跟我说了下他的解法，我说原来就是用两个heap，他说是的。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315215&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一题大概就是找二叉树里面节点值相同的连续序列的最大长度，这个序列只能是从父节点到一个子节点，然后一直往下走。这个题目很简单，直接递归就做出来了。

```

1
3      2
1 1    1 1

```

比如这样一个二叉数，最大的长度是3.

第二个题目是第一个题目的follow up。没有只能是父节点到一个子节点的约束，可以横向。比如上面这个树中最长序列的长度是4。

这一问其实也不难，同样直接递归就可以了。在LC做过类似的题目，而且面试之前看面经的时候看过这个题目。  
所以我就也没思考太多，直接写代码了。结果却因为自己没有定义好递归函数的输入和返回，在写递归函数的边写边在脑子想递归的输入输出。

结果就出错了，递归函数里面逻辑写得不清楚。最后再面试官的帮助下，才写完了代码，还有很多bug。所以当时面完感觉很不好。

```
int longestPath(TreeNode *root) {
    if(root == nullptr) return 0;

    int left = 0, right = 0;
    if(root->left != nullptr && root->val == root->left->val) {
        left = longestPath(root->left);
    }
    if(root->right != nullptr && root->val == root->right->val) {
        right = longestPath(root->right);
    }

    int res = max(left, right) + 1;
    return res;
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315555&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

第一轮：东欧口音大叔

1. 给一个数，问是不是两个完全平方数的和，用2sum的方法解，讨论了test case、复杂度以及 $i * i \leq n$ 这种写法不好，没有考虑到溢出

```
bool isSquare(int n) {
    if(n < 0) return false;
    return pow(int(sqrt(n)), 2) == n;
}

bool twoSquare(int n) {
    for(int i = 1; i < ceil(sqrt(n)); i++) {
        if(isSquare(n - i * i)) return true;
    }
    return false;
}
```

2. 给一个matrix，0和1组成，问有没有长方形（4个顶点都是1）。我后来发现这是一个fb的面经，不过当时我是先暴力解，再一步步优化的，每一步的思路面试官都要求我打出来。当然啦大叔也非常nice，优化到他觉得可以了才写code，最后时间不够还差一点，他说没关系，让我pseudocode把剩下的两三行完成了。

第二轮：ABC小哥，刚入职15个月

## 里特扣 而起就 改版 要求返回结果

这题我大概两个月前写过，只知道要用dp，面试时候基本是现写的，后来我看了答案，果然和我写的完全不一样。不知道是不是我真的有点慢，还是因为我的方法和答案差太多，小哥总是打断我，问这问那，搞得我非常着急，慌慌忙忙地被催着写完了。之后小哥让我设计了很多test case，口头跑test case我竟然也没检查出问题，最后在他的提醒下修了两个bug，感觉自己傻透了。最后分析了时空复杂度。

面完我感觉很差，第二轮这么简单的题我还做成这样，和他的交流也感觉很紧张很不顺，面试的节奏完全被带跑了。  
焦急地等过了圣诞和新年，果然在2号一大早收到了加面通知，还好还好，还有机会。

1/10 加面：中国小姐姐，入职3年半

## 里特扣 伞伞起 改版 n-ary tree

很顺利地bug free写完，口头跑了test case，分析了复杂度，小姐姐表示都很好。还剩十几分钟她就开始和我闲扯了，让我不要担心实习转正的面试，又说害怕我match组有点晚，当时觉得小姐姐真是太好了！

3, 4, 2, 1, 5, X

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=315666&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

国人大哥（叔）

1. 从双链表里删除第一个匹配的值，跟上：删除所有匹配的值

```
void deleteInList(ListNode* head, int val) {
    ListNode *curr = head;

    ListNode *prev, *next;
    while(curr != nullptr) {
        prev = curr->prev;
        next = curr->next;
        if(curr->val == val) {
            if(prev != nullptr) prev->next = curr->next;
            if(next != nullptr) next->prev = curr->prev;
            delete curr;
        }
        curr = next;
    }
}
```

2. 有一个函数 f：如果是奇数就乘三加一；如果是偶数就除以二。重复apply函数f直到变为一为止。写一个程序计算需要多少次循环变为1。跟上：如果输入是一个数列，对每一个元素计算变为一的次数。

```
int f(int a) {
    return a % 2 == 0 ? a / 2 : a * 3 + 1;
}

vector<int> count(vector<int> &a) {
    vector<int> res(a.size());
```

```
unordered_map<int, int> dict;

for(int i = 0; i < a.size(); i++) {
    int tmpCount = 0;
    int curr = a[i];
    while(curr != 1) {
        if(dict.find(curr) != dict.end())
            tmpCount += dict[curr];
        break;
    }
    curr = f(curr);
    tmpCount++;
}
dict[a[i]] = tmpCount;
res[i] = tmpCount;
}

return res;
}
```

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=316555&extra=page%3D1%26filter%3Dsorid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

## 第一轮：

1. 写一个函数，随机等概率返回矩形中的一个点，输入输出参数自己定义
  2. 现有一系列矩形（无重叠），随机等概率返回这些矩形中的一个点，可以用第一个函数
  3. 若矩形有重叠，重写第二题的函数

## 第二轮：

1. 如下图，○表示建筑物，×表示空，最下横线表示地面，去掉建筑物的某一块，求问那些部分会掉落

00000

XOXXX

XOXXX

xooxx

— — — — —

2 著

23133

x4xxx

xyxxxx

x9xxx

10

- 加面：

  1. 蠢口而溜溜
  2. 给一系列sorted的Interval，无重叠，写三种insert和find函数，要求查找的时间复杂度分别为 $O(n)$ ,  $O(\log n)$ ,  $O(1)$   
(面试的时候让说了3种思路，就让写了 $O(\log n)$ 的代码)

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=316425&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sorti>

## d%3D311

第一轮：美国小哥，给了一个数据库，里面有所有兔子及其他们之间亲属关系的信息，给你个函数getParents() 可以得到某一只兔子的父母，input是某一只兔子，你需要返回这只兔子所有的祖先。一点点确定题目的条件，比如返回值，祖先有没有重复。。然后小哥问如何找两只兔子的最近的共同祖先，可以利用之前一问写的函数，不过貌似我当时想的不好，写完代码时间差不多也到了，后来想想好像是有更好的办法。小哥基本什么问题都会先来一句“That's a(an) interesting/good question. What do you think about it.”,对了，最后还讨论了下时间复杂度和set的时间复杂度（因为上面的返回值最后定的是set）

第二轮：亚洲小哥？看起来就像个geek，问的是地里见过的实现一个Map(key, value, currentTime + ttl) 含有 get() 和 put(), 会有个getTime() 可以得到当前时间，两个Map分别存value 和 time 就好，后面一问是如果需要你删除过期的数据，用priorityqueue替代存time的map效率会比较高

lunch：一个伯克利的美国小哥，貌似我不问问题，他就不说话，所以只有不停尬聊，人还可以，免费的午餐跟我在某a实习吃的付费的差不多...

第三轮：跟另一个面试的人，房间冲突，面试过程中还有两三轮敲门，然后更没想到的是第三轮的面试官，估计是一欧洲大叔？他一开口我就知道跪了，听不懂说话，还好第一问说是 warmup question很简单，就是两个本来相同String，有一个插入了一个字符，找出它， $O(n)$  解法很简单，然后followup是要你效率高一点，我说二分法吧，还没想到怎么做，后来想说可以在每一步二分的时候，向左向右匹配，然后他给我举了个例子：aaaaaaaaaa，插入的也是a，我就...，然后只有继续想，最奔溃的是，最后大叔估计也急了，给我了提示，结果说的就是我上面那个方法，我说这个办法不是不好么？他说极端情况下是不好啊，但是遇到正常的英语单词就会快不少，不是？然后最后9分钟，硬着头皮开始写，还在揪二分的细节的时候，时间到了。可能当初他给我举那个例子只是想确认我有没有考虑到，然而我以为是否认了这个方法，交流听不懂，这个教训有点大。

第四轮：一个有点白头发的亚洲人，人看起来就很好，跟他记忆最深的是好能聊，就一题，find the minimum difference of BST, 比如1,5,6,9，返回1( 6-5 )，确认好条件之后，他硬是让我口述了一遍才让开始写代码，然后问了时间空间复杂度，基本所有的选择都会问你为什么，写完之后，花了很长时间聊这题用stack怎么做，举了个例子，模拟了一遍，还争了下细节，他还说“你放心，这个部分不算在面试里面，你要是发现我错了，可以发信给hr,让她转给我”...后来因为是最后一轮，结束聊了好多，刚开始还因为上一轮的情况有点紧张，他说，我也还依然记得当初我没做出来的那题，说完之后想了下，恩确实记得，23333，算是开导我硅谷机会多，一次没做出来没什么，最后送我到打uber的地方。对了 我还特地问了有没有followup，竟然是真的没有。

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=316626&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

1. 机器人扫地的题。只给出机器人的move , rotate , clean函数，让扫整个屋子。此题上来说dfs的算法，但是真正实现起来好多细节要理清。

2轮， 给一个treenode的数组，问这些node是不是能组成一颗树。就是一棵树的所有点都要在数组里，数组不能包含多余的点，并且只能是一棵树。如果有多个root就不算。

3轮， 里的口 兮零児， followup，如果是多叉树怎么办

4轮， hangman的猜字游戏。给一串word，和一个secret word，让你猜出这个词是啥。你说一个字母，然后会告诉你不在这个词里，比如猜banana，你猜a，会返回结果 -a-a-a，让你如何以最优方式去猜这个词。如果挂，就挂这一题上。我先猜一个出现在最多words里的字母，根据结果，从list里删除那些不符合结果的词。可惜只写了一半。这位白人大叔一直引导，

却总get不到他的点。

```
struct TrieNode {
    unordered_map<char, TrieNode*> children;
    bool isWord;
    int deep;
}

TrieNode() {
    this->isWord = false;
    this->deep = -1;
}
};

class Trie {
    TrieNode *root;
    string answer;

void buildTree(vector<string> &words, int n) {
    root = new TrieNode();
    answer = string(n, '_');

    for(auto word : words) {
        addWord(word);
    }
}

void addWord(const string &word) {
    TrieNode *curr = root;

    for(auto c : word) {
        if(curr->children.find(c) == curr->children.end()) {
            curr->children[c] = new TrieNode();
        }
        curr->children[c]->deep = curr->deep + 1;
        curr = curr->children[c];
    }
    curr->isWord = true;
}

bool guess(TrieNode *node, int i, vector<char> &path) {
    if(i == len) result.push_back(path);
    if(answer[i] != '_') return guess(node->children[answer[i]], i + 1);
    bool res = false;

    for(auto child : node->children) {
        vector<int> idx = foo(child.first);
        saveIdx(idx);
        fillAnswer(idx);
    }
}
```

```

    res = res || guess(child, i + 1);

    loadIdx();
}
return res;
}

void fillAnswer(vector<int> &idx) {}
vector<int> foo(char c) {return vector<int>();}
};

```

5轮，在一个有序数组里找一个数出现的次数。比如【1, 2, 3, 3, 3, 3, 4, 4, 5】，3出现的次数是4次，4出现的是两次。二分查找。半小时做完。小哥用多余的时间出了一道system design的题，说里个大概，并没有参考价值，就不说了。

觉得出了第四轮，其他应该都给出答案来，3, 5可能会strong hire，多谢国人哥哥姐姐！如果跪，就是第四轮了。。。不知道如果有一个no hire是不是就完全悲剧了。

现场有chromebook可以用，（建议用）这样写出来会比我的白板至少好看不少。

祝大家好运。骑驴找马甚是辛苦，加油！

<http://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=316970&extra=page%3D1%26filter%3Dsortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311>

运气肥肠肥肠好，coding部分水果，背靠背。两个面试官都是白人而且一个组的，第一个是leader听口音年纪稍长，第二个是同组手下干活的小哥。

第一面：维护一个排序的double linked list，完成insert方法。followup是再写一个方法进行排序，merge sort就可以。

第二面：简单但是比较tricky的题目，交换箱子游戏，给一堆箱子，每个都有编号，比如：3, 4, 2, 1, 5, X

X代表空位，如何用最少的次数交换箱子使得箱子有序：1,2,3,4,5,X 不能直接交换两个箱子，需要借助空位X

跟这个题很像：<https://stackoverflow.com/questions/...-with-minimum-moves>

optimized algo是需要 n + 1步



