

友塔游戏 offer

面试岗位：平台后台开发

2022/2/11 投完简历就发了笔试链接

2022/2/11 笔试

3道题，可能我抽的比较简单，一个模拟，一个dfs，一个二分，大概40分钟全AC了

2022/2/15 一面 (30m) 电话

问题：

1. 回顾笔试的三道算法题，怎么做的
2. DFS算法的过程
3. 口撕算法：合并K个有序数组
4. 口撕算法：TOPK问题
5. 口撕算法：逆波兰表达式是啥知道不，如何构造
6. 动态规划的经典问题有啥，一个01背包的转移方程怎么来的
7. 二维的dp有哪些
8. 快排最坏情况的比较次数，归并排序呢，知道归并吗
9. 为什么想到学Go
10. 简历上Go项目的简单描述
11. 简历上安卓项目的简单描述，为啥用Kotlin
12. 你觉得Go语言为什么会发展出来
13. 反问

感受：

体验不错，春招第一次面试，个人有点紧张，不过问题都不算很难，主要考察算法和数据结构，没有问八股，面完也直接告诉我等二面通知，晚上就和hr约好了

2022/2/18 二面 (1h)

问题：

1. 口撕算法：[772. 基本计算器 III](#)，请口述算法执行过程
2. 为什么学Go
3. 为什么用Gin，跟Django开发的区别
4. Go和Python的差异
5. Go没有类有什么影响
6. Go的Redirect是怎么做到的，重定向的原理（响应头自带的location）
7. 缓存击穿，MySQL如何应对（锁机制）
8. 个人职业规划，如何看待技术和业务的关系
9. 反问

感受：

面试官很好，个人回答不好，算法是会写的，但是口述过程没说出来，前面说逆波兰表达式还给自己挖了坑，面试官会一步步引导

2022/2/18 hr面

晚上hr打电话

2022/2/21 Offer

流程真的很快，其他公司连笔试的消息都没有，这边offer都给了，可以考虑15天

字节跳动

面试岗位：后台开发-基础架构

2022/2/25 hr电话

捞面，没有笔试，说当天面，没时间又约定面试时间

2022/2/28 一面（1h）凉

忘记录音了。。只能靠回忆了，所以问题不全

首先用5分钟自我介绍+项目介绍

问题：

1. https，端口，证书怎么验证的（回答了ssl协商过程），追问协商具体原理（不知道）（此时被面试官说网络知识不太好，心态逐渐失衡）
2. http状态码，501、502？（只知道500、503）
3. base62是啥，为什么有base64
4. mysql 联合索引abc，要我手写联合索引的sql查询（联合索引不知道是啥，寄）
5. Redis怎么使用IO（I/O多路复用，多嘴提了用select、kqueue等库）
6. kqueue是什么（不知道，我说只知道select、poll、epoll）
7. select、poll、epoll 的区别
8. 信号和信号量是什么，信号量如何使用
9. kill -2 和 kill -9的区别（只晓得 -9 和 -15）
10. 数学题：n米，每次走的距离递增（1米，2米，...），每次可向前或向后，最少多少正好走n米
11. 算法题：LRU缓存，增加了个生存时间，x秒后自动过期（自动过期不需要写，只在查询的时候判断，可以判断经过的秒数是否超过生成时间来淘汰）
12. 反问

感受：

第一次面大厂，刚开始表现的很紧张，回答问题时说了一大堆，怕自己说的不具体，被面试官提示要说重点，感觉语气有点不耐烦了。。

反思：

被刷可能是项目相关的没答出来和数学题没给出正解，比如只知道base62和base64的区别，但是没有关注base64编码主要作用，然后数据库索引查询我是真的没接触。。

关于那道数学题，我后面想到的是：

转化为： $1+2+\dots+x = n$ ，即Leetcode目标和那道题

$$\begin{aligned} left - right &= n \\ right + left &= (1 + x) * x / 2 \\ left &= [(1 + x) * x + 2n] / 4 \end{aligned}$$

然后遍历x+01背包

后面看了貌似有数学公式可以直接算出来。。。

2022/3/23 飞书后端捞

2022/3/30 一面 (40min)

问题：

1. Redis的基本结构，怎么对key设置过期时间
2. 一段Go的代码，问输出啥（主要考察defer后进先出）

```
1 func test() {
2     x := 1
3     defer func() {
4         fmt.Printf("first %d\n", x)
5     }()
6     x = 2
7     defer func() {
8         fmt.Printf("second %d\n", x)
9     }()
10    defer func() {
11        fmt.Printf("third %d\n", x)
12    }()
13 }
14 // third 2
15 // second 2
16 // first 2
```

3. 有缓冲channel和无缓冲channel的区别
4. 进程间的通信方式，哪个效率最高，有名管道和无名管道的区别
5. 什么是僵尸进程，僵尸进程太多对系统的影响
6. 如何让进程后台运行
7. TCP如何保证可靠传输，拥塞控制的细节
8. UDP报文的最大报文限制（说错了，应该是DNS中UDP限制512字节）
9. 联合索引，慢查询怎么定位
10. 数据库的隔离级别，都讲讲
11. 对称加密和非对称加密的区别，哈希散列算不算加密算法

12. 状态码, HTTP2.0

13. 手撕算法:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  /*
5   一个智能水龙头, 可以设置指定时间切换水流速度 (比如从第m秒开始的水流速度为n毫升/秒), 计算接满一桶
   水需要的时间;
6   示例输入:
7   水流速度设置: [{0,10}, {5,20}, {20,15},
8   {10,30}], 每个pair对{m,n}的m表示时间 (秒, 非负整数), n表示水流速度 (毫升/秒, 正整数)
9   桶容量: 1000 (毫升, 正整数)
10  示例输出: 56.67 (秒)
11  */
12
13  struct cmp {
14      bool operator()(pair<int, int>& x, pair<int, int>& y) {
15          return x.first > y.first;
16      }
17  };
18
19  double count(int n, vector<pair<int, int>>& speed) {
20      priority_queue<pair<int, int>, vector<pair<int, int>>, cmp> que;
21      for (auto s : speed) que.push(s);
22      double ans = 0;
23      auto s = que.top();
24      que.pop();
25      while (n && !que.empty()) {
26          auto t = que.top();
27          que.pop();
28          if (s.second * (t.first - s.first) <= n) {
29              ans += t.first - s.first;
30              n -= s.second * (t.first - s.first);
31          } else {
32              ans += n * 1.0 / s.second;
33              n = 0;
34          }
35          s = t;
36      }
37      if (n) ans += n * 1.0 / s.second;
38      return ans;
39  }
40
41  int main() {
42      ios::sync_with_stdio(false);
43      cin.tie(0);
44      vector<pair<int, int>> speed;
45      speed.push_back(pair<int, int>{0, 10});
```

```

46     speed.push_back(pair<int, int>{5, 20});
47     speed.push_back(pair<int, int>{20, 15});
48     speed.push_back(pair<int, int>{10, 30});
49     cout << count(2000, speed);
50     return 0;
51 }

```

14. 反问

感受：

比上次友好太多了，问的偏基础，算法也不太难

2022/4/6 二面（45min）凉

问题：

1. 说一个项目，问了用的base62库怎么生成的，为什么这样写
2. 布隆过滤器
3. redis的expire如何实现的，数据结构有哪些，讲讲其中一个（说了zset跳表），追问跳表插入删除的过程
4. slice和map的底层结构，slice的append操作扩容，slice作切片操作还是原来的吗，为什么slice不初始化也能用（不会）
5. Go的GPM调度，哪些场景会触发协程调度
6. Go的垃圾回收，从根开始遍历，根是怎么决定的，除了三色标记还要哪些GC方法
7. 事务隔离级别，底层怎么实现的，隔离级别有什么用，给了代码不同隔离级别下的输出结果
8. 删除一个目录下所有.c的文件，怎么删除
9. HTTPS怎么实现的，SSL协商过程
10. 手撕代码：[K 个一组翻转链表](#)
11. 反问

感受：

纯背八股没啥用呀，再问往下问一层就不会了。。基本每个问题都问了底层怎么实现的。。。。

手撕代码一行写错，没能跑通样例，还是不能去字节咯

深信服

面试岗位：Golang开发

2022/3/2 笔试

go基础知识的选择和填空，两道算法：一道编辑距离，一道岛屿问题变版，30分钟不到全AC了。

2022/3/8 一面 (30m) 凉

问题：

1. slice和数组的区别
2. slice指向数组的指针什么时候会变
3. map的底层原理
4. map是并发安全的吗，怎么解决，sync.map怎么实现的
5. linux常见文件系统（不了解）
6. 项目中自己做了什么，项目做完有什么提升
7. 手撕算法：合并两个有序数组
8. 博客都是自己写的吗（多看博客整理笔记）
9. 为什么看《Redis的设计与实现》（博客有）
10. 蓝桥杯是什么（尬住了😓）
11. 手撕算法：反转链表（那个编译器不知道什么版本，C++建的结构体编译错误，我后面本地运行没问题）
12. 大学期间哪个课程学的好（计网），以太网基于什么协议（😓😓，我说应用层传输层多），HTTP滑动窗口的原理
13. TCP如何保证可靠传输
14. 反问

感受：

八股回答的自我感觉还行，手撕算法也出来了，第二天感谢信了😁

虾皮

面试岗位：后端开发

2022/3/7 笔试

算法题：2.9/3

```
1  class Solution3 { // 这题只过了90%
2      private:
3          long long ans = INT64_MAX;
4          void dfs(long long x, long long y, long long targetX, long long targetY,
5                  long long cnt) {
6              if (x > targetX || y > targetY || cnt >= ans ) return;
7              if (x == targetX && y == targetY) {
8                  ans = min(ans, cnt);
9                  return;
10             }
11             if (x <= targetX - 1 && y <= targetY - 1)
12                 dfs(x + 1, y + 1, targetX, targetY, cnt + 1);
13             if (x > INT64_MAX / 2 || y > INT64_MAX / 2) return;
14             if (x * 2 <= targetX && y * 2 <= targetY)
15                 dfs(x * 2, y * 2, targetX, targetY, cnt + 1);
16         }
17     }
```

```

18     public:
19         /**
20          * Note: 类名、方法名、参数名已经指定, 请勿修改
21          *
22          *
23          * 将 sourceX, sourceY 转换成 targetX, targetY 最少需要多少次计算
24          * @param sourceX long长整型 x初始值
25          * @param sourceY long长整型 y初始值
26          * @param targetX long长整型 x目标值
27          * @param targetY long长整型 y目标值
28          * @return long长整型
29          */
30         long long GetMinCalculateCount(long long sourceX, long long sourceY,
31                                         long long targetX, long long targetY) {
32             if (sourceX > targetX || sourceY > targetY) return -1;
33             dfs(sourceX, sourceY, targetX, targetY, 0);
34             if (ans == INT64_MAX) return -1;
35             return ans;
36         }
37     };
38
39     class Solution2 {
40     public:
41         /**
42          * Note: 类名、方法名、参数名已经指定, 请勿修改
43          *
44          *
45          * 计算用户逾期扣分数
46          * @param dpdInfo string字符串 用户逾期记录, 'N': 没有逾期, 正常
47          * 'Y': 有逾期
48          * @return int整型
49          */
50         int calDPDScore(string dpdInfo) {
51             int cnt = 0;
52             int mc = 0;
53             for (char c : dpdInfo) {
54                 if (c == 'Y')
55                     cnt++;
56                 else
57                     cnt = 0;
58                 mc = max(cnt, mc);
59             }
60             if (mc == 0) return 0;
61             if (mc > 0 && mc <= 3) return -10;
62             if (mc > 3 && mc <= 7) return -15;
63             if (mc > 7) return -25;
64         }
65     };
66
67     class Solution1 {

```

```

68     public:
69         /**
70          * Note: 类名、方法名、参数名已经指定, 请勿修改
71          *
72          *
73          * 找到数组中乘积最大的连续子数组, 并返回乘积
74          * @param nums long长整型 vector 原始数组
75          * @return long长整型
76          */
77         long long GetSubArrayMaxProduct(vector<long>& nums) {
78             int n = nums.size();
79             if (n == 0) return 0;
80             if (n == 1) return nums[0];
81             long long x = 0, y = 0, nx, ny; // x正, y负
82             if (nums[0] >= 0)
83                 x = nums[0];
84             else
85                 y = nums[0];
86             long long ans = nums[0];
87             bool tag = false;
88             for (int i = 1; i < n; i++) {
89                 if (nums[i] == 0) {
90                     nx = 0;
91                     ny = 0;
92                 } else if (nums[i] > 0) {
93                     nx = (x != 0) ? x * nums[i] : nums[i];
94                     ny = y * nums[i];
95                 } else {
96                     nx = y * nums[i];
97                     ny = (x != 0) ? x * nums[i] : nums[i];
98                 }
99                 ans = max({ans, nx, ny});
100                 x = nx;
101                 y = ny;
102             }
103             return ans;
104         }
105     };

```

2022/3/18 感谢信

大概没后端HC了, 免笔试转推到了测试和运维, 拒了

花旗金融

投递岗位: 花旗消费者银行技术部 GCT

2022/3/8 笔试

选择+三道算法题，第一题getline（字符串含空格）读入，第二题最大连续子数组和，第三题字符运算

最后一题本地运行给我电脑卡死机了。。。后面写出来了

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  // 3*[a2*[c]] => accaccacc
4  stack<string> strs;
5  stack<char> ops;
6  int main() {
7      // ios::sync_with_stdio(false);
8      // cin.tie(0);
9      // freopen("input.txt", "rt", stdin);
10     // freopen("output.txt", "wt", stdout);
11     string s;
12     cin >> s;
13     int n = s.size();
14     int j, k, t;
15     for (int i = n - 1; i >= 0; i--) {
16         if (s[i] == ']')
17             ops.push(s[i]);
18         else if (s[i] == '[')
19             ops.pop();
20         else if (s[i] == '*') {
21             j = i - 1;
22             while (j >= 0 && s[j] >= '0' && s[j] <= '9') j--;
23             t = 0;
24             for (k = j + 1; k < i; k++) t = t * 10 + (s[k] - '0');
25             string tmp = strs.top();
26             strs.pop();
27             string ans = "";
28             while (t--) ans += tmp;
29             while (j >= 0 && s[j] != '[') ans = s[j--] + ans;
30             strs.push(ans);
31             i = j;
32         } else if (!ops.empty()) {
33             string tmp = "";
34             j = i;
35             while (j >= 0 && s[j] != '[') {
36                 tmp = s[j--] + tmp;
37             }
38             strs.push(tmp);
39             i = j;
40         }
41     }
42     cout << strs.top();
43     return 0;
```

2022/3/15 一面 (30min)

英文自我介绍和一些英文问答，5个面试官轮流问，这阵容让人害怕

问题：

1. 简历上项目介绍（英文）
2. Go和Python的区别（英文）
3. 项目里的荣誉奖项，项目相关问题
4. 前端怎么设计的，对前端有什么更深入的看法（没深入学过前端）
5. 大学期间的专业课，C++比起Java的优势
6. 短网址项目，为什么要短网址服务
7. 其他个人相关问题
8. 反问

感受：

可能是我不会Java，感觉技术相关没怎么问，大部分都是我的简历和项目经历吧，感觉这个部门比较偏向前端。。

后面一直没消息了。。。

OPPO offer

面试岗位：安卓开发

投早了，后面看到有后端开发岗位。。啥都没准备直接面了，居然一路畅通，只能说无心栽柳柳成荫。。。

2022/3/4 笔试

Java八股+Java限定算法题，给我这个不会Java的整懵逼了。。。全靠代码补全写出来，题不难，全AC了

2022/3/9 一面 (15min)

问题：

1. 说一个比较熟悉的开源项目，说一下原理（说了gin）
2. kotlin的协程和Java线程的区别，线程调度（主要想问调度）
3. OKHttp的原理（不了解，只知道使用）
4. 项目做什么，难点，怎么解决
5. Java HashMap原理（不会Java）
6. 中间人攻击的原理（不了解，只知道中间人攻击是什么）
7. HTTPS和HTTP的区别，证书校验
8. 三次握手，四次挥手
9. Kotlin的优点
10. 有用过Android Studio吗
11. 反问（此时面试官委婉地提出要我好好学学协程，意思很明显了🤔）

感受：

不会Java很掉分，不过面试体验很好，没有感觉到难受，聊的很开心。

2022/3/17 二面 (30min)

问题：

1. 说一个比较满意的项目，做了什么工作，结果是啥
2. 说说 Jetpack Compose，和传统开发的优势，怎么学习的
3. 其他Jetpack了解吗（说了Room和Datastore的使用），Room注解运行时的情况（不了解）
4. 怎么学Kotlin的，协程了解吗，描述一下
5. 为什么想到学Android项目
6. 项目持续时间很长，在这段时间做了什么
7. 基本数据结构有哪些，它们的特点
8. OKHTTP的功能，哪些地方应用，做了哪些优化
9. 大量网络请求，怎么优化（不会，应该是在客户端做离散化处理）
10. 短网址项目，Redis是怎么学习的
11. 在学校，其余时间是怎么安排的，加了什么社团吗（我说去飞盘社扔飞盘，面试官笑了。。。)
12. 课程成绩相关，最喜欢哪个学科
13. 考研之类的问题
14. 反问

感受：

面试体验很好，都是正常的聊聊天，主要问的都是我做了什么，面试官最后反问环节给我提了很多实用的建议，以后有时间一个试试吧

面试最后直接告诉我等hr沟通了

2022/3/18 hr面 (30min)

看牛客说hr面刷人不少，不过我面了30分钟感觉挺有诚意的。

三个优点，缺点，期望薪资，为什么选base地之类的，期待offer

2022/3/22 Offer

进入录用环节辽

美团

面试岗位：后端开发

2022/3/5 笔试凉

凉凉，全是ACM题，5道前2道卡63%，后面直接放弃，第一题题目理解错了，签到都没成功。。

这难度是我不配了，没想到下一周全是easy middle题。。。写早了。。。

一直泡池子卡在笔试那，听同学说美团不裁员，但是非TOP985基本都会卡

携程

面试岗位：Java后端开发

2022/3/10 笔试

4道编程，2.5/4

2022/3/18 一面 (30min)

问题：

1. Java的顶层父类（不会Java。。。）
2. 讲讲Go中的常用结构
3. 无缓冲channel没有其他协程消费，写入后会阻塞吗
4. Go的值传递和引用传递
5. Go中实现同步的三种工具（说了锁、channel）
6. 了解哪些数据库（Redis和Mysql），SQL和NOSQL的区别
7. Redis的数据结构，为什么用跳表，跳表怎么跳的，一定是2个吗（应该是随机化的）
8. 为什么Redis数据结构设计的这么精巧（引导我说是内存占用小，空间复杂度低）
9. Redis为什么用单线程，Redis的数据结构是线程安全的吗
10. 关系型数据库事务的特性，如何实现的，redo为什么不直接写脏页（应该是顺序写和随机写）
11. 同时查询数据库，数据一定一样吗（引导说了MVCC）
12. 了解的常用数据结构，树一般用来干嘛，对堆的理解，堆的作用（说了排序，这里想要是维护最大最小值，退化一点）
13. 反问：对后端接触较少，有什么建议

感受：

面试官说科班应该更好的掌握基础，深以为然了，很多都是硬背八股，被暴打了

问的都是偏底层偏原理，我一直在那会不会复读，给我整不好意思了都。。只背八股是不行的，得好好了解原理和实现，以后慢慢补充吧。不过面试体验非常好，面试官会引导说出正确答案，也会跟我解释，奈何自己太菜🤦

2022/3/28 二面 (35min) 凉

问题：

1. 对面向对象的理解，面向对象解决了什么问题
2. 有没有研究什么分布式的框架（没），讲一下用Go做的挑战最大的项目
3. 短链接访问失效怎么解决，过期之后怎么处理
4. Redis的set，redis的多线程模型
5. Innodb的锁
6. Mysql数据量过大有什么方法解决（分库分表），多表的聚合查询怎么操作
7. Mysql的B+树，B+树比起B树的优势（提示I/O方面的减少）
8. 设计模式（单例、工厂），单例有几种，区别，单例多线程怎么处理，单例解决了什么问题（提示稀有资源的创建）
9. python学了多久（用来爬虫），反爬机制了解吗
10. 手撕算法：跟团游，如何规划行程使得行程距离最短（说了爆搜、dp、dijkstra和bellman-ford算法）
11. 反问

感受：

面试体验挺好的，给了很多建议，回头查了一些没回答出来的问题，很多都是Java框架方面的解答，比如Mysql分库分表聚合查询是通过ES实现的。。感觉我和携程技术栈不太匹配吧，面完没多久就进人才池了。。。

BIGO

投递岗位：Go开发（工程效率）

2022/3/22 一面（45min）

问题：

1. 进程和线程的区别，进程通信除了常规的还有其他吗
2. 协程了解吗，为什么比线程更轻量，协程堆栈是怎么管理的
3. Go同步、异步、阻塞、非阻塞的区别，阻塞和非阻塞IO呢（不了解）
4. 多线程怎么保证线程安全，Go的读写锁，两个读可以吗，一直读加锁，写会不会得不到锁（会，但没思考过怎么解决）
5. 网络用过epoll模式吗（没用过）
6. C++map的实现（说了红黑树和哈希），如何考虑选择，为什么用红黑树
7. TCP如何保证可靠性，拥塞控制（说了四种方法）
8. 动态链接和静态链接的区别（不了解）
9. 说说短链接项目，碰撞问题怎么解决的，用什么数据结构存的，如何实现短时间内同一个长链生成同一个短链（我说了用set存一个长链-短链的映射关系）
10. Redis其他数据结构，zset的实现
11. 说说安卓项目
12. 手撕算法：有序数组中某个数出现的次数（二分呗）
13. 有一个1GB大小的文件，文件里每一行是一个词，每个词的大小不超过16B，内存大小限制是1MB，要求返回频数最高的100个词(Top 100)(之前没遇到过，说了分治读取，哈希统计，大顶堆获得top100，和面试官探讨了很久)
14. HTTPS如何安全（老八股了）
15. 反问：工程效率部门主要业务
16. 反问：对我的建议，评价还好，和面试官随便聊了聊

感受：

面试感觉挺好，问问题，聊聊天

2022/3/29 二面（55min）

问题：

1. 短网址为什么要做9位短码，在redis里会占多少空间（开头一击，感觉不妙），怎么实现的，怎么去重，两个线程生成同一个短码怎么解决（提示Redis的事务，不会）
2. 多线程环境下对变量的读写操作是否是原子的（提示CAS，说了概念），追问内部怎么实现的（不会）
3. 协程怎么实现并发的，怎么做系统I/O的
4. 阻塞I/O和非阻塞I/O，编程里怎么用（没用过）
5. epoll用来干啥的，写过epoll吗（没写过）
6. 静态链接和动态链接
7. 打开一个网址，发生了什么，HTTP报文怎么发送Server的（提示路由，不会）

8. 如果你要做一个路由器，怎么设计路由表，知道目标IP怎么找到下一跳地址（没看这方面，我都傻了，后面一直追问）
9. 函数调用寄存器和栈是怎么变的（提示BP/EBP，汇编，不会）
10. Linux的Fork怎么实现的（不会）
11. 虚拟地址和物理地址怎么管理
12. 页表是啥，页多大，为啥4KB
13. 手撕算法：[117. 填充每个节点的下一个右侧节点指针 II](#)（写了层序遍历，追问空间 $O(1)$ 的方法，说了跟着next走，用递归写的没写出来，后面看迭代更好写。。。)
14. 反问，回答的很敷衍，感觉凉凉

感受：

emmmmm，问的是不是有点奇怪？寄了

很长时间没感谢信，查询还在面试流程，泡着了

哔哩哔哩

投递岗位：后台开发

2022/3/31 一面

拒了，有点后悔没参加了

华为

投递岗位：应用软件开发

2022/3/30 笔试

1.4/3，第3题不知道怎么读入一棵树，看评论好像是读一个字符串😂，第二天发了性格测评，后面发了是否参加面试，然后一直没消息

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  // we have defined the necessary header files here for this problem.
5  // If additional header files are needed in your program, please import here.
6  int m,n,sx,sy,ex,ey;
7  int k;
8  int ans=0;
9  // 卡 40%
10 bool check(int x, int y){
11     if (x<0 || y<0 || x>=m || y>=n) return false;
12     return true;
13 }
14
15 void dfs(vector<vector<int>>& dp, int target, int cnt, int x, int y){
16     if (!check(x,y) || cnt!=dp[x][y]) return;
17     if (cnt==target){
```

```

18         if (x==ex && y==ey) ans++;
19         return;
20     }
21     dfs(dp,target,cnt+1,x+1,y);
22     dfs(dp,target,cnt+1,x,y+1);
23     dfs(dp,target,cnt+1,x-1,y);
24     dfs(dp,target,cnt+1,x,y-1);
25 }
26
27
28
29 int main()
30 {
31     // please define the C++ input here. For example: int a,b; cin>>a>>b;;
32     // please finish the function body here.
33     // please define the C++ output here. For example:cout<<____<<endl;
34     cin>>m>>n;
35     cin>>sx>>sy>>ex>>ey>>k;
36     vector<vector<int>> dp(m,vector<int>(n,-1));
37     while (k--){
38         int x,y;
39         cin>>x>>y;
40         dp[x][y]=INT32_MAX;
41     }
42     if (!check(sx,sy) || !check(ex,ey)){
43         cout<<0<<" "<<0;
44         return 0;
45     }
46     if (sx==ex && sy==ey){
47         cout<<1<<" "<<0;
48         return 0;
49     }
50     dp[sx][sy] = 0;
51     int cnt = 1;
52     for (int i=sx+1; i<m; i++) dp[i][sy] = cnt++;
53     cnt = 1;
54     for (int j=sy+1; j<n; j++) dp[sx][j] = cnt++;
55     cnt = 1;
56     for (int i=0; i<sx; i++) dp[i][sy] = cnt++;
57     cnt = 1;
58     for (int j=0; j<sy; j++) dp[sx][j] = cnt++;
59     for (int i=sx+1; i<m; i++){
60         for (int j=sy+1; j<n; j++){
61             if (dp[i][j]==INT32_MAX) continue;
62             if (dp[i][j-1]==INT32_MAX && dp[i-1][j]==INT32_MAX) dp[i][j] =
INT32_MAX;
63             else dp[i][j] = min(dp[i-1][j],dp[i][j-1])+1;
64         }
65         for (int j=sy-1; j>=0; j--){
66             if (dp[i][j]==INT32_MAX) continue;

```

```

67         if (dp[i][j+1]==INT32_MAX && dp[i-1][j]==INT32_MAX) dp[i][j] =
        INT32_MAX;
68         else dp[i][j] = min(dp[i-1][j],dp[i][j+1])+1;
69     }
70 }
71 for (int i=sx-1; i>=0; i--){
72     for (int j=sy+1; j<n; j++){
73         if (dp[i][j]==INT32_MAX) continue;
74         if (dp[i][j-1]==INT32_MAX && dp[i+1][j]==INT32_MAX) dp[i][j] =
        INT32_MAX;
75         else dp[i][j] = min(dp[i+1][j],dp[i][j-1])+1;
76     }
77     for (int j=sy-1; j>=0; j--){
78         if (dp[i][j]==INT32_MAX) continue;
79         if (dp[i][j+1]==INT32_MAX && dp[i+1][j]==INT32_MAX) dp[i][j] =
        INT32_MAX;
80         else dp[i][j] = min(dp[i+1][j],dp[i][j+1])+1;
81     }
82 }
83 dfs(dp,dp[ex][ey],0,sx,sy);
84 cout<<"ans<<" "<<dp[ex][ey];
85 return 0;
86 }

```

小黑盒

投递岗位：后端开发

2022/3/16 笔试

30分钟3道编程，2.2/3，卡的那题有思路了但是没时间写了。。。

2022/3/21 一面

拒了

小米

投递岗位：Golang开发

2022/3/24 一面

拒了

库洛游戏

投递岗位：游戏服务器后台开发

2022/3/23 笔试

拒了

叠纸游戏

投递岗位：游戏服务端开发

2022/3/30 笔试

和华为重复了，拒了

米哈游 简历挂

投递岗位：后台开发

鹰角网络 简历挂

投递岗位：后端开发

网易雷火 简历挂

投递岗位：Web后台开发

京东 无岗位

投递岗位：Go开发

裂开，爆出裁员之后，这个岗位都没了

SHEIN 简历挂

投递岗位：后台开发

商汤 没反应

投递岗位：后台开发

腾讯音乐 TME 没反应

投递岗位：后台开发

