

返回列表

1

2

1 / 2 页


下一页

查看: 1197 | 回复: 19

[面试经验] 亚麻 OA2 超级详细

[复制链接]

试试Instant~

 jinwei14 发表于 2018-12-30 01:38:58 | 只看该作者

垅头 电梯直达

本楼: **【顶】** 100% (3)

0% (0) **【踩】**

全局: 顶 100% (53)

0% (0) 踩

2018(10-12月) **码农类General** **硕士** **全职@Amazon** - 校园招聘会 - **在线笔试** | **Other** | fresh grad应届毕业生

希望看到这篇文章的朋友记住经验教训，都是泪啊。

亚麻OA2 for new grad 120 min workflow 加 70 分钟写代码。不是社招！. From 1point 3acres bbs

一般题目只会出

以下内容需要积分高于 130 您已经可以浏览

. 1point3acres

1. Count number of substrings with exactly k distinct characters

2. MaximumMinimumPath (DFS)

3. subtree: maximum average node.

注意: : : ! ! ! ! ! 用python 的朋友, python 用的是 2.7! ! ! ! ! ! ! ! ! ! !

. From 1point 3acres bbs

workflow 看下边: 地里捞的, link 也有:

以下内容需要积分高于 130 您已经可以浏览

https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=468231

1/11

0.0 --work simulation:

..用户和deadline第一

..<https://www.1point3acres.com/bbs ... 26orderby%3Ddatelin>

e

..<https://www.1point3acres.com/bbs ... %3Dradio&page=1>

. check 1point3acres for more.

..screen shot

<https://www.1point3acres.com/bbs/thread-448373-1-1.html>

..

<https://www.1point3acres.com/bbs ...hlight=amazon%2Boa2>

Work simulation(原则有先后顺序)

目前两大做题中最重要的原则:

- 1.requirement排在第一, deadline第二。
- 2.有manager出现的选项无脑选manager, manager就是一个组的地头蛇。

Amazon9条主要原则

原则1: 客户是上帝, requirement优先, 任何影响上帝的事情都不能干,

如某个requirement影响了上帝的体验,

你就是死键盘上也不能砍了, 宁愿miss deadline

原则2: 为长远考虑, 即客户几年之后可能会出现的需求也要考虑到,

不会为了交付短期的deadline,

而牺牲长期的价值。(比如 [global api](#) 和 local api)

原则3: 最高标准, “最高”对应上面的“长远”。

原则4: 一般情况, 能请示manager就请示manager, manager一般不会出错

原则5: 速度很重要, 决策和行动都可以改变, 因此不需要进行过于广泛的推敲. 1point3acres

, 但提倡在深思熟虑下进行冒险。

原则6: 不需要一定要坚持“非我发明”, 需求帮助也是可以的, 四处寻找创意

, 并且接受长期被误导的可能. From 1point 3acres bbs

原则7: 敢于承担责任, 任劳任怨, 比如领导说谁会java, 你会你就跳出来说我。 From 1point 3acres bbs

原则8: 对问题刨根问底, 探究细节

原则9: 服从大局 (团队比个人重要)

打分不是关键, 排序才是关键。

大部分情况下其实并没有deadline 和 requirement谁更好, 更多还是在这个组合中你对ddl 和 requirement整体的权衡。

每个选项可以评1~5分, most effective 是5, 然后1是least effective

刚开始让你看一些介绍amazon工作环境的视频

1.上来给一段video, 场景是项目的晨会, 就是把team正在推进的项目描述一下,

期间会有多个项目和你有关系, 后面会遇到

2.进入工作界面, 可以看到接受到邮件, 接收到的instant message

3.进入工作状态。会有同事给你发邮件, 发信息。需要你对他们提出的问题做一些判断, 也就是给解决问题的选项评分

4.一个21题, 有[log分析bug](#), 有给报告出问题结论, 有判断项目 走向的

情境1: 给图书馆写图书推荐系统, 关于book api

两个人, 在表达不同的观点

选择: [tell me more](#)

一开始其实每个人都在强调自己是对的, 即使有一个人更对一些,

也应该选tell me more (原则8), 选了之后会得到更多信息. 1point3acres

情境2: 选图书馆的服务器有没有开放关于实体书的api

两个小哥讨论图书推荐的api应该是自己做还是用现成的。

[自己做api覆盖面广](#), 但是due赶不上, 别人做的能赶上due。

requirement优先 (原则2), [tell me more层层递进](#)

. 1point3acres

情境3: 经理说咱们最近服务器老挂, 什么情况?

先看internal bug的记录

选 I think service 3 is the problem,



but I would like to see another report to confirm
烙印，义正辞严说自己做了20年服务器，不可能有错误，
刚刚调试过服务器，不可能是内部错误。
[选自己去查](#)，问题的关键在于不要麻烦别人
增加开发过程中测试的时间/[测试覆盖更多case](#)，放5
[写Manuel test](#)，放3
[还有个是unit test](#)，也放3
增加QA的人手，放1
让客户来当小白鼠发现问题，放1

情境4: Amazon recommendation system item，
给你推荐一些你感兴趣的item，第一个issue总是失败，
[第二个issue总是显示germany](#)
第一个问题是因为username 太长所以一直报错。
第二个问题是因为[他用proxy的name来决定是不是语言了](#)。

情境5: 德国amazon除了什么问题，让你看log回答问题。问你大概哪里除了
问题
亚马逊推荐广告，给英国人推了德文广告，给你log文件，
问你可能在哪儿，找bug in error log

情境6: 员工们讨论case media network服务器最近好多complaints
有德国的，有invalid recommendation，有返回404，
找出错误原因的相同点
德语因为服务器，一个因为用户名太长，一个是有些用户的语言变成德语

情境7: 具体客户ddl 只有两周，两个方案，延到四周，做完整。 . 1point3acres
另一个说先实现一部分功能做个demo，再慢慢做。
先做demo放5，按部就班四周放3， 通知其他组说两走做不完接着做美国放1

情境8: 估计项目开发时间. check 1point3acres for more.
Manager放5，找有经验的人请教4，上网查资料或是先做一段时间再估计都
放3，
还有其他裸上的就1。

情境9: 一个项目时间表设计
说你是这里最会用什么语言的，比如java

情境10: 安排会议
视频会议 5 三个老二开会和老二去找老大开会 3 推迟会议和邮件开会 1

情境11: 搞个数据库
两周时间可以搞个数据，**ben可以帮忙，大腿priya可以帮，但是要等一周
半
报告manager放5，和**合作等大腿放4，合作/等大腿是3
自己单干，cut feaure都是1

情境12: 系统是否升级
做两个feature，一个让100%用户爽，一个让20%用户爽，
但要升级系统，升级系统自己组会爽，但是升级会推迟做的feature，
不升级吧，升级之后还得做一遍
这题的中心是不升级，先做feature，先让用户爽。
先做100的feature再升级，再做20的feature，放5
不升级，因为我们承诺要做feature，放4。
不升级，要搞定feature，可以以后推了其他ddl再升级，放3
不升级，因为对其他组没影响，我们应该focus在request上面，放2
升级，推迟这两个feature的ddl，因为升级造福子孙后代，放2
升级，不然要做两次，放1
这题的关键在于升不升级，要坚定的站在一边

情境13: 新产品设计
给8周时间，选择题，让你pick up 一个features的组合要求利益最大化，
每个feature都有相应的价值，H >> M >> L 都代表远大于
首先ddl是前提，中位数不能超过8太多，那样的话就算feature再多也没意
义，
同价值，按照ddl排序，同ddl按照价值排序。

情境17: 代码分析

三段一长选最长

下边是三道题的解答：不一定全对但是答题思路一样：..表示摘抄自地里的大佬的总结。

以下内容需要积分高于 188 您已经可以浏览

1. Count number of substrings with exactly k distinct characters

..给一个字符串，求满足条件的substring的数量：substring有刚好K个distinct characters

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

..最多含k个字母的substr

..k distinct character：我就直接用的hashset存字符的 $O(n^2)$ 解法，可以过test case，但是需要注意的是，如果 $k=0$ ，需要返回0，我当时一直以空字符串也属于答案，返回1。。被这个case卡了40分钟。。

..利口散司令 找所有的substring最多重复k个character

..substring with k distinct characters 给字符串和k，找子串，子串里面的不同的字母有k个，问有多少这样的子串

..the number of the substring with exactly k distinct character.

我用的two for loop with hashmap. 因为写完时间还有太多，我又试了试用two pointer，但是好像并不行，但是我最后也并没有搞懂为什么two pointer不行。查了查好像最优的就是 $O(n^2)$ 。不知道地里大神有没有高见。我总是觉得好像能搞一个 $O(n)$ 的出来。

..coding一道是k distinct substring, 类似鑫口159, 但千万注意返回是一共有多少这样的子串，而不是找其中最长的一个

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

class Solution:

. 1point3acres

slicing windows

def findSubstring(self, str, k):

if str is None or len(str) == 0:
return 0

ch_count = {}

n = len(str)
start, end = 0, 0

count = 0 - baidu 1point3acres

while end < n:

ch_count[str[end]] = ch_count.get(str[end], 0) + 1

if len(ch_count) == k:

count += self.getCount(end, str, ch_count)

ch_count[str[start]] -= 1

if ch_count[str[start]] == 0:

del ch_count[str[start]] 1point3acres

start += 1

if len(ch_count) < k:

end += 1

```
return count

def getCount(self, end, str, ch_count):
    n = len(str)
    count = 0
    while end < n:
        if str[end] not in ch_count:
            break
        count += 1
        end += 1

    return count
```

2. MaximumMinimumPath (DFS)

(一个小哥去爬山，给一个矩阵，要从左上角爬到右下角，每一步只能往右往下走。矩阵的每一个值是这一点海拔。显然有很多条路可以走，每一条路都有一个经过的海拔最低值，求所有路径的海拔最低值的最大值。

比如：

5 4 5

1 3 6

有三条路可以走：5 1 3 6，5 4 3 6，5 4 5 6。三条路的最小值分别是1,3,4。求的就是这三个数的最大值，也就是4。我直接暴力写的。)

..给一个二维矩阵，表示某点的高度，人从左上角出发，只能向下或向右走，最终到达右下角。

求所有可能的路径中每个路径最低点的最大值。

典型DP，类似LC174

好像没在地里见过。

..maximum minimum path：直接二维dp做了，没啥特别奇怪的地方。。

..<https://www.1point3acres.com/bbs ... p;page=1#pid5030286>

..Max min path from top left to bottom right.

基本的DP，开始用了个matrix存，后来有时间就优化了一下用个1D array存。肯定是最优了。

-baidu 1point3acres

..另一道是miximin path，经典的dp问题。两道题的Java代码都可以在小土刀上找到。

..2) maximum minimum path，可以参考这个帖子：

<http://www.1point3acres.com/bbs/thread-443128-1-1.html>。我准备时正好没有写到这个题，

只好硬写，用了DFS遍历。我用python写的，然后看到IDE里提供的API没有定义class

(不是像利扣的那样有定义个 class Solution(object))，

所以没法用def __init__(self)来定义 self.maximum 这个变量(因为需要不断更新)。

我后来有些急，就用了一个minimumValueList 来存每条路径的最小值，最后返回 max(minimumValueList)。但这样的复杂度其实很高，后来OA结束后，我重写了一遍，用 global maximum 就可以了。

总结来说，还是自己对 global 的使用不太熟。

```
class Solution3:
```

```
def findMaxMin(self, matrix):
```

```
    self.max_val = -sys.maxsize - 1
    self.row = len(matrix)
    self.col = len(matrix[0])
    min_val = sys.maxsize
```

```
    self.dfs(matrix, min_val, 0, 0)
```

. 1point3acres

```
    return self.max_val
```

```
def dfs(self, matrix, min_val, i, j):
```

-baidu 1point3acres

```
    if i >= self.row or j >= self.col:
```

```
        return
```

```
    if i == self.row - 1 and j == self.col - 1:
```

```
        min_val = min(min_val, matrix[j])
```

```
        self.max_val = max(self.max_val, min_val)
```

```
        return
```

```
    min_val = min(min_val, matrix[j])
```

```
    self.dfs(matrix, min_val, i, j + 1) -baidu 1point3acres
```

```
    self.dfs(matrix, min_val, i + 1, j)
```

3. subtree: maximum average node (<https://yeqiuquan.blogspot.com/2013/04/maximum-average-node.html>)

..多叉树最大子树平均。地里有，我的变种是平均不带自己。比如树

```
4
```

```
/ | \
```

```
1 2 3
```

这样的情况4这个node的平均是 $(1+2+3)/3$ 而不是 $(1+2+3+4)/4$

Tree里面找所有node平均值最大的子树，输出root。（leave node不算子树。）

..结点+他的子树/结点个数的最大值。第二题我用的bottom up，迷之test case有几个一直过不了，又看不到test case内容，又不能自己定义test case来测试

..最大子tree node值的平均值。在北美读的大学实在不知道怎么用中文描述请谅解。注意不是binary tree。其实直接backtrack就好。

..愚蠢的想问一下第二题的testcase 1:

```
1->2->3
```

```
expected answer: 2
```

为什么不是3 因为除去leaf node

..maximum average subtree，这个tree是N-ary tree。

求每个node以及其所有后代的value平均值，返回平均值最大的node

..2. maximum average subtree

<https://yeqiuquan.blogspot.com/2013/04/maximum-average-node.html>

我照着这个链接的方法写的

真正题目里面每个node有好几个子节点. 1point3acres

需要注意叶子结点不能作为结果

..第二题最近高频的n叉树找最大的sum/node数的结点，dfs就完了，注意要

排除叶子结点。

Definition for a Node.

class Node:

```
def __init__(self, val, children):
    self.val = val
    self.children = children
```

class Solution:

```
def mas(self, root):
    if root is None:
        return None
```

```
self.max_average = -sys.maxsize
self.node = None
self.find_max_average(root)
```

```
return self.node
```

```
def find_max_average(self, root):
```

```
if len(root.children) == 0:
    # if root.val > self.max_average:
    #     self.max_average = root.val
    #     self.node = root
    return 1, root.val. From 1point3acres bbs
```

```
count, sum = 0, 0
```

```
for child in root.children:
```

```
    c_count, c_sum = self.find_max_average(child)
    count += c_count
    sum += c_sum
```

```
if sum / count > self.max_average:
    self.max_average = sum / count
    self.node = root
```

```
return sum, count. check 1point3acres for more.
```

. From 1point3acres bbs

包含所有有用信息的传送门:

<https://www.1point3acres.com/bbs ... 6orderby%3Ddateline>
小土刀! <https://wdxub.com/interview/14520850399861.html>

. check 1point3acres for more.

. From 1point3acres bbs

#在这里快速回复#

○ 评分

上一篇: [面试经验]谷歌电面2018年11月底
下一篇: 12/29 狗家OA 第一题变了

○ 本帖被以下淘专辑推荐:

· 2018招聘 | 主题: 150, 订阅: 14

· 亚麻New Grad | 主题: 29, 订阅: 2