Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by 李鹏辉，元培学院

说明：

1）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

2）提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

3）如果不能在截止前提交作业，请写明原因。

编程环境

Windows 10 Home, PyCharm 2022.3.2 (Community Edition)

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

28170: 算鹰

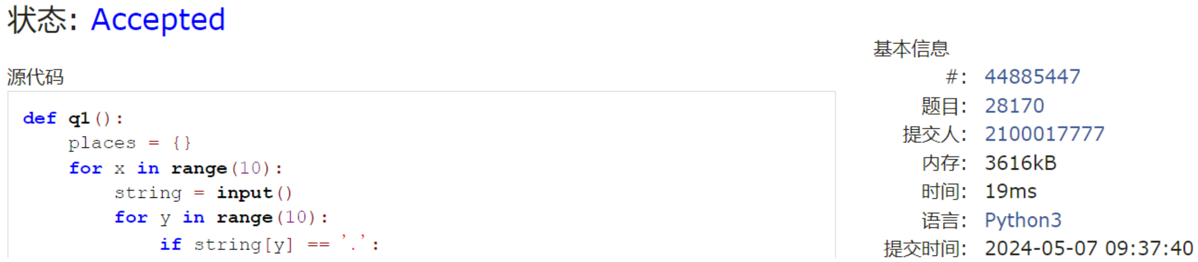
dfs, http://cs101.openjudge.cn/practice/28170/

思路：约30分钟。一开始理解错题意了，题目的表述有歧义而且恰巧按照歧义理解样例输入可以产生样例输出。

代码

```python
def q1():
    places = {}
    for x in range(10):
        string = input()
        for y in range(10):
            if string[y] == '.':
                places[(x, y)] = ''

    result = 0
    visited = {}

    def helper(x, y):
        visited[(x, y)] = ''
        moves = [(-1, 0), (1, 0), (0, 1), (0, -1)]
        for dx, dy in moves:
            new_place = (x+dx, y+dy)
            if new_place in places and new_place not in visited:
                helper(x+dx, y+dy)

    for x, y in places.keys():
        if (x, y) not in visited:
            helper(x, y)
            result += 1

    print(result)
    return


q1()
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```python
def q1():
    places = {}
    for x in range(10):
        string = input()
        for y in range(10):
            if string[y] == '.':
```

基本信息

#:	44885447
题目:	28170
提交人:	2100017777
内存:	3616kB
时间:	19ms
语言:	Python3
提交时间:	2024-05-07 09:37:40

02754: 八皇后

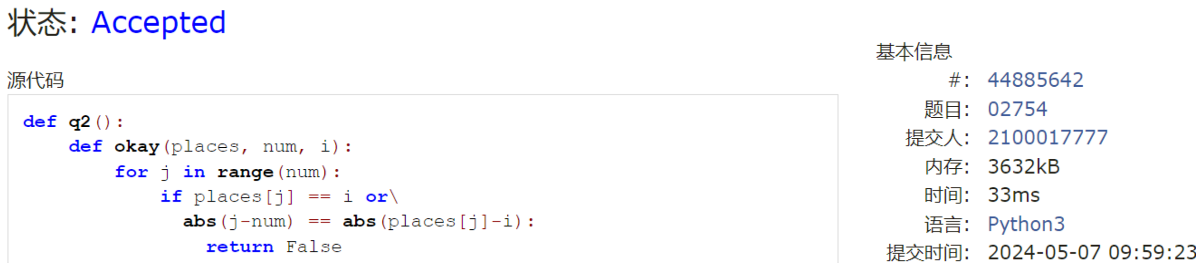
dfs, http://cs101.openjudge.cn/practice/02754/

思路：约23分钟。

代码

```python
def q2():
    def okay(places, num, i):
        for j in range(num):
            if places[j] == i or\
              abs(j-num) == abs(places[j]-i):
                return False
        return True

    s = [[[i+1] + [-1]*7, 1] for i in range(8)]
    re = []
    while s:
        places, num = s.pop()
        if num == 8:
            re.append(''.join(map(str, places)))
        else:
            for i in range(1, 9):
                if okay(places, num, i):
                    np = places[:]
                    np[num] = i
                    s.append([np, num+1])

    re = list(map(int, re))
    re.sort()
    for _ in range(int(input())):
        i = int(input())
        print(re[i-1])
    return


q2()
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
def q2():
    def okay(places, num, i):
        for j in range(num):
            if places[j] == i or\
              abs(j-num) == abs(places[j]-i):
                return False
```

基本信息

#: 44885642
题目: 02754
提交人: 2100017777
内存: 3632kB
时间: 33ms
语言: Python3
提交时间: 2024-05-07 09:59:23

03151: Pots

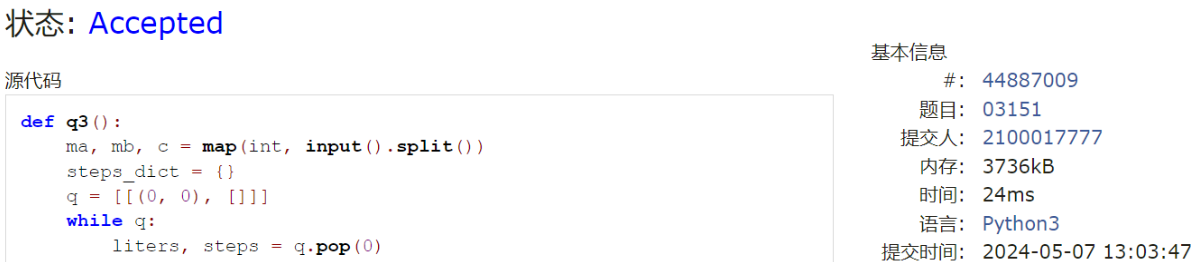
bfs, http://cs101.openjudge.cn/practice/03151/

思路：约25分钟。

代码

```
def q3():
    ma, mb, c = map(int, input().split())
    steps_dict = {}
    q = [[(0, 0), []]]
    while q:
        liters, steps = q.pop(0)
        a, b = liters
        if a == c or b == c:
            print(len(steps))
            [print(step) for step in steps]
            return
        if liters in steps_dict:
            continue
        steps_dict[liters] = steps
        if a < ma:
            steps.append('FILL(1)')
            q.append([(ma, b), steps[:]])
            steps.pop()
            if b > 0:
                steps.append('POUR(2,1)')
                if ma - a > b:
                    q.append([(a+b, 0), steps[:]])
                else:
                    q.append([(ma, a+b-ma), steps[:]])
                steps.pop()
        if b < mb:
            steps.append('FILL(2)')
            q.append([(a, mb), steps[:]])
            steps.pop()
            if a > 0:
                steps.append('POUR(1,2)')
                if mb - b > a:
                    q.append([(0, a + b), steps[:]])
                else:
                    q.append([(a + b - mb, mb), steps[:]])
                steps.pop()
        if a > 0:
            steps.append('DROP(1)')
            q.append([(0, b), steps[:]])
            steps.pop()
        if b > 0:
            steps.append('DROP(2)')
            q.append([(a, 0), steps[:]])
            steps.pop()
    print('impossible')
    return


q3()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
def q3():
    ma, mb, c = map(int, input().split())
    steps_dict = {}
    q = [[(0, 0), []]]
    while q:
        liters, steps = q.pop(0)
```

基本信息

#:	44887009
题目:	03151
提交人:	2100017777
内存:	3736kB
时间:	24ms
语言:	Python3
提交时间:	2024-05-07 13:03:47

05907: 二叉树的操作

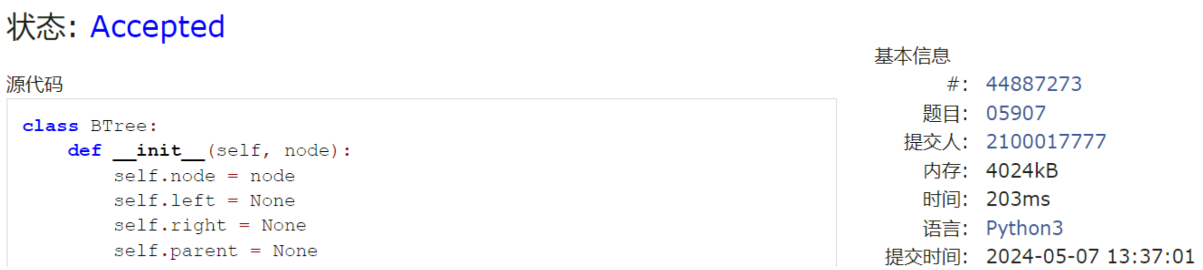
http://cs101.openjudge.cn/practice/05907/

思路：约30分钟。

代码

```python
class BTree:
    def __init__(self, node):
        self.node = node
        self.left = None
        self.right = None
        self.parent = None


def q4():
    for _ in range(int(input())):
        n, m = map(int, input().split())
        nodes = [BTree(i) for i in range(n)]
        for _ in range(n):
            node, leftnum, rightnum = map(int, input().split())
            if leftnum != -1:
                nodes[node].left = nodes[leftnum]
                nodes[leftnum].parent = nodes[node]
            if rightnum != -1:
                nodes[node].right = nodes[rightnum]
                nodes[rightnum].parent = nodes[node]
        t = nodes[0]

        for _ in range(m):
            nums = list(map(int, input().split()))
            if nums[0] == 2:
                ct = nodes[nums[1]]
                while ct.left is not None:
                    ct = ct.left
                print(ct.node)
            if nums[0] == 1:
                a, b = nums[1:]
                a_parent = nodes[a].parent
                b_parent = nodes[b].parent
                a_dir = 'left' if a_parent.left == nodes[a] else 'right'
                b_dir = 'left' if b_parent.left == nodes[b] else 'right'
                exec('a_parent.' + a_dir + ' = nodes[b]')
                exec('b_parent.' + b_dir + ' = nodes[a]')
                nodes[a].parent = b_parent
                nodes[b].parent = a_parent
    return


q4()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
class BTree:
    def __init__(self, node):
        self.node = node
        self.left = None
        self.right = None
        self.parent = None
```

基本信息

#:	44887273
题目:	05907
提交人:	2100017777
内存:	4024kB
时间:	203ms
语言:	Python3
提交时间:	2024-05-07 13:37:01

18250: 冰阔落 I

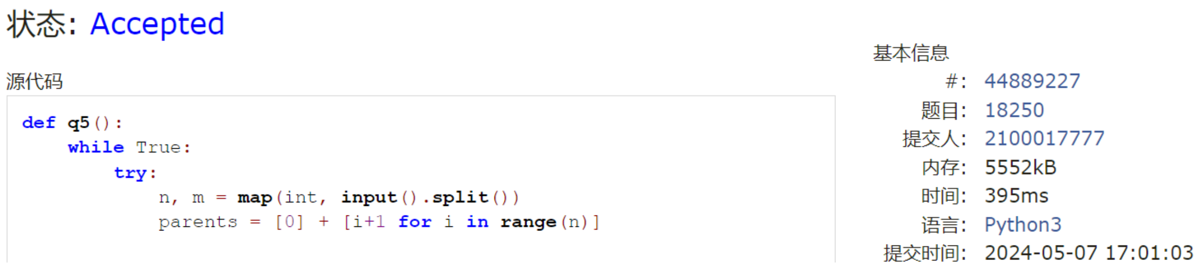
Disjoint set, http://cs101.openjudge.cn/practice/18250/

思路：约40分钟。

代码

```python
def q5():
    while True:
        try:
            n, m = map(int, input().split())
            parents = [0] + [i+1 for i in range(n)]

            def find(a):
                if parents[a] == a:
                    return a
                else:
                    parents[a] = find(parents[a])
                    return parents[a]

            for _ in range(m):
                a, b = map(int, input().split())
                if find(a) == find(b):
                    print('Yes')
                else:
                    print('No')
                    parents[find(b)] = a

            left = []
            for i in range(1, n+1):
                root = find(i)
                if root not in left:
                    left.append(root)
            left.sort()
            print(len(left))
            print(' '.join(map(str, left)))
        except:
            return


q5()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```python
def q5():
    while True:
        try:
            n, m = map(int, input().split())
            parents = [0] + [i+1 for i in range(n)]
```

基本信息

#:	44889227
题目:	18250
提交人:	2100017777
内存:	5552kB
时间:	395ms
语言:	Python3
提交时间:	2024-05-07 17:01:03

05443: 兔子与樱花

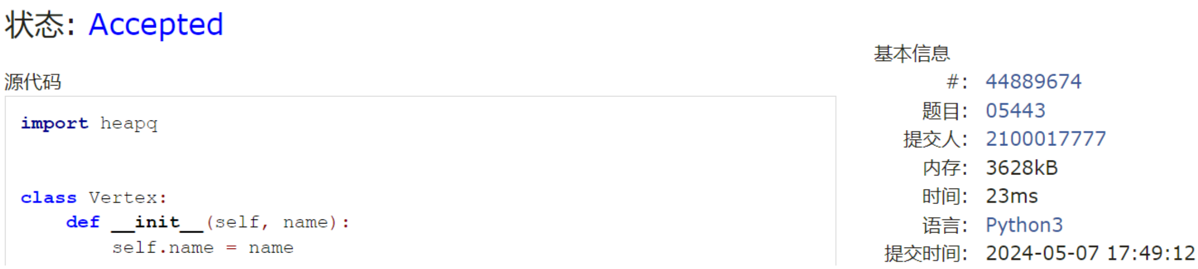
http://cs101.openjudge.cn/practice/05443/

思路：约40分钟。

代码

```python
import heapq


class Vertex:
    def __init__(self, name):
        self.name = name
        self.connected_to = {}
        self.previous = None
        self.distance = 9999999999


class Graph:
    def __init__(self):
        self.vertices = {}

    def get(self, name):
        return self.vertices[name]

    def add_edge(self, f, t, weight):
        self.get(f).connected_to[self.get(t)] = weight
        self.get(t).connected_to[self.get(f)] = weight


def q6():
    g = Graph()
    n = int(input())
    for _ in range(n):
            name = input()
            g.vertices[name] = Vertex(name)
        n = int(input())
        for _ in range(n):
            a, b, d = input().split()
            d = int(d)
            g.add_edge(a, b, d)
        n = int(input())
        for _ in range(n):
            s, e = input().split()
            c = g.get(s)
            c.distance = 0
            pq = []
            heapq.heappush(pq, (0, c))
            while True:
                if not pq: break
                cd, c = heapq.heappop(pq)
                if c.name == e: break
                for np, d in c.connected_to.items():
                    if cd + d < np.distance:
                        np.distance = cd + d
                        np.previous = c
                        heapq.heappush(pq, (np.distance, np))
            re = [e]
            while c != g.get(s):
                last = c.previous
                re.append(f'({c.connected_to[last]})')
                re.append(last.name)
                c = last
            print('->'.join(re[::-1]))
            for vertex in g.vertices.values():
                vertex.previous = None
                vertex.distance = 9999999999
        return


q6()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```python
import heapq


class Vertex:
    def __init__(self, name):
        self.name = name
```

基本信息

#:	44889674
题目:	05443
提交人:	2100017777
内存:	3628kB
时间:	23ms
语言:	Python3
提交时间:	2024-05-07 17:49:12

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

可以用exec函数加上命令的字符串省去一点分类讨论的步骤。