

assignmentB

Assignment #B : 图 (1/4)

Updated 2031 GMT+8 Nov 17, 2025

2025 fall, Complied by 杨知进 物理学院

说明:

1. 解题与记录:

对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. 提交安排：提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
3. 延迟提交：如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E07218: 献给阿尔吉侬的花束

bfs, <http://cs101.openjudge.cn/practice/07218/>

思路：

好久没写bfs，动手一下漏洞百出，再次温习。

代码：

```
from collections import deque

dx = [0, 0, -1, 1]
dy = [1, -1, 0, 0]

def bfs(graph, r, c, start, end):
```

```

visited = [[0]*c for _ in range(r)]
queue = deque()
queue.append((start[0], start[1], 0))
visited[start[0]][start[1]] = 1
while queue:
    x, y, step = queue.popleft()
    if (x, y) == end:
        return step
    for i in range(4):
        nx, ny = x+dx[i], y+dy[i]
        if 0<=nx<r and 0<=ny<c and not visited[nx][ny] and graph[nx][ny] != '#':
            visited[nx][ny] = 1
            queue.append((nx, ny, step+1))
else:
    return -1

n = int(input())

for i in range(n):
    graph = []
    r, c = map(int, input().split())
    start, end = 0, 0
    for j in range(r):
        row = input().strip()
        graph.append(row)
        for x in range(c):
            if row[x] == 'S':
                start = (j, x)
            elif row[x] == 'E':
                end = (j, x)
                break
    result = bfs(graph, r, c, start, end)
    if result == -1:
        print('oop!')
    else:
        print(result)

```

代码运行截图 (至少包含有"Accepted")

#50984591提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```
from collections import deque
```

基本信息

#: 50984591
题目: 07218
提交人: 25n2400011575

用时：30min左右

M27925: 小组队列

dict, queue, <http://cs101.openjudge.cn/practice/27925/>

思路：

实质上队列是属于小组的队列，在大queue中只要排小组编号（deque），teamqueue中排内部编号（dict: int→deque）。

代码：

```
from collections import deque
import sys
t = int(input())
dic = {}
for i in range(t):
    persons = list(map(int, input().split()))
    for person in persons:
        dic[person] = i
teamqueue = {}
queue = deque()

for line in sys.stdin:
    command = line.strip()
    if command == 'STOP':
        break
    elif command == 'DEQUEUE':
        if queue:
            firstteam = queue[0]
            if firstteam in teamqueue and teamqueue[firstteam]:
                person = teamqueue[firstteam].popleft()
                print(person)
                if not teamqueue[firstteam]:
                    queue.popleft()
    else:
        x = int(command.split()[1])
        if x in dic:
            team = dic[x]
            if team in teamqueue and teamqueue[team]:
                teamqueue[team].append(x)
            else:
                queue.append(team)
```

```

        if team not in teamqueue:
            teamqueue[team] = deque()
            teamqueue[team].append(x)
    else:
        id = -x
        queue.append(id)
        if id not in teamqueue:
            teamqueue[id] = deque()
            teamqueue[id].append(x)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
import sys
```

基本信息

#: 50984885

题目: 27925

提交人: 25n2400011575

image-2.png

用时: 30min左右

M04089: 电话号码

trie, <http://cs101.openjudge.cn/practice/04089/>

思路:

代码:

```

class Node:
    def __init__(self):
        self.child = {}
        self.end = 0

def insert(root, s):
    cur = root
    for c in s:
        if cur.end:
            return 0
        if c not in cur.child:
            cur.child[c] = Node()
        cur = cur.child[c]
    if cur.end:
        return 0
    if cur.child:
        return 0
    cur.end = 1

```

```

return 1

t = int(input().strip())
for i in range(t):
    n = int(input().strip())
    nums = [input().strip() for j in range(n)]
    nums.sort()
    root = Node()
    ok = 1
    for num in nums:
        if not insert(root, num):
            ok = 0
            break
    print("YES" if ok else "NO")

```

代码运行截图 (至少包含有"Accepted")

#50984972提交状态

[查看](#) [提交](#)

状态: Accepted

基本信息

#: 50984972

题目: 04089

源代码

image-3.png

用时: 15min左右

M3532.针对图的路径存在性查询I

disjoint set, <https://leetcode.cn/problems/path-existence-queries-in-a-graph-i/>

思路:

并查集。

代码

```

class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int,
    queries: List[List[int]]) -> List[bool]:
        id = [0] * n
        for i in range(1, n):
            id[i] = id[i - 1]
            if nums[i] - nums[i - 1] > maxDiff:
                id[i] += 1
        return [id[u] == id[v] for u, v in queries]

```

代码运行截图 (至少包含有"Accepted")

Python3 ▾ 智能模式

```
1 class Solution:
2     def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries: List[List[int]]) -> List[bool]:
3         id = [0] * n
4         for i in range(1, n):
5             id[i] = id[i - 1]
6             if nums[i] - nums[i - 1] > maxDiff:
7                 id[i] += 1
8         return [id[u] == id[v] for u, v in queries]
```

已存储

题目描述 | 通过 × | 题解 | 提交记录

← 全部提交记录

通过 550 / 550 个通过的测试用例

题解

image-4.png

用时：20min左右

M19943: 图的拉普拉斯矩阵

OOP, graph, implementation, <http://cs101.openjudge.cn/pctbook/E19943/>

要求创建Graph, Vertex两个类，建图实现。

思路：

代码

代码运行截图 (至少包含有"Accepted")

T25353: 排队

<http://cs101.openjudge.cn/pctbook/T25353/>

思路：

代码：

代码运行截图 (至少包含有"Accepted")

2. 学习总结和个人收获

这次作业回过头运用了很多之前的算法，也跟ai讨论了一下bfs如何能实现一次得到最短等以前只是接受和忽略的问题。