Assignment #8: 图论: 概念、遍历,及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by <mark>李鹏辉,元培学院</mark>

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

19943: 图的拉普拉斯矩阵

matrices, http://cs101.openjudge.cn/practice/19943/

请定义Vertex类, Graph类, 然后实现

思路:约40min.

```
class Vertex:
def __init__(self, key):
    self.id = key
    self.connected_to = {}

def add_neighbor(self, nbr, weight=0):
    self.connected_to[nbr] = weight
```

```
class Graph:
10
11
        def __init__(self):
             self.vert_list = {}
12
             self.num_vertices = 0
13
14
15
         def add_vertex(self, key):
             self.vert_list[key] = Vertex(key)
16
             self.num_vertices += 1
17
18
19
         def __contains__(self, n):
             return n in self.vert_list
20
21
         def add_edge(self, f, t, weight=0):
22
23
             if f not in self.vert_list:
                 self.add_vertex(f)
24
25
             if t not in self.vert_list:
                 self.add_vertex(t)
26
             self.vert_list[f].add_neighbor(self.vert_list[t], weight)
27
28
29
        def __iter__(self):
30
             return iter(self.vert_list.values())
31
32
33
    def q1():
34
         def 11m(n, edges):
35
36
             graph = Graph()
37
             for i in range(n):
38
                 graph.add_vertex(i)
39
             for edge in edges:
                 a, b = edge
40
41
                 graph.add_edge(a, b)
                 graph.add_edge(b, a)
42
43
             matrix = []
             for vertex in graph:
44
                 row = [0] * n
45
46
                 row[vertex.id] = len(vertex.connected_to)
47
                 for neighbor in vertex.connected_to.keys():
                     row[neighbor.id] = -1
48
49
                 matrix.append(row)
50
             return matrix
51
        m, n = map(int, input().split())
52
53
         edges = []
54
         for i in range(n):
             a, b = map(int, input().split())
55
56
             edges.append((a, b))
57
58
         re_matrix = 11m(m, edges)
         for row in re_matrix:
59
             print(' '.join(map(str, row)))
60
61
62
63
    q1()
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

```
      源代码
      #: 44668178

      class Vertex:
      题目: 19943

      def __init__(self, key):
      提交人: 2100017777

      self.id = key
      内存: 3716kB

      self.connected_to = {}
      时间: 29ms

      def add_neighbor(self, nbr, weight=0):
      提交时间: 2024-04-15 21:27:36
```

基本信息

18160: 最大连通域面积

matrix/dfs similar, http://cs101.openjudge.cn/practice/18160

思路:约30min.

```
class Vertex:
1
 2
        def __init__(self, key):
 3
            self.id = key
 4
            self.connected_to = {}
 5
        def add_neighbor(self, nbr, weight=0):
 6
 7
            self.connected_to[nbr] = weight
8
9
10
    class Graph:
        def __init__(self):
11
            self.vert_list = {}
12
13
            self.num_vertices = 0
14
15
        def add_vertex(self, key):
            self.vert_list[key] = Vertex(key)
16
17
            self.num_vertices += 1
18
19
        def __contains__(self, n):
20
            return n in self.vert_list
21
        def add_edge(self, f, t, weight=0):
22
            if f not in self.vert_list:
23
                 self.add_vertex(f)
24
            if t not in self.vert_list:
25
                 self.add_vertex(t)
26
27
            self.vert_list[f].add_neighbor(self.vert_list[t], weight)
28
        def __iter__(self):
29
            return iter(self.vert_list.values())
30
31
32
```

```
def q2():
33
34
35
         def dfs(graph, vertex, visited):
             visited[vertex.id] = True
36
37
             area = 1
38
             for neighbor in vertex.connected_to:
39
                 if not visited[neighbor.id]:
                     area += dfs(graph, neighbor, visited)
40
             return area
41
42
         t = int(input())
43
44
         for _ in range(t):
             m, n = map(int, input().split())
45
             graph = Graph()
46
             grid = []
47
48
             # add vertices
             for i in range(m):
49
                 row = input().strip()
50
                 grid.append(row)
51
52
                 for j in range(n):
53
                     if row[j] == 'W':
54
                          graph.add_vertex((i, j))
55
             # add edges
56
57
             for i in range(m):
58
                 for j in range(n):
59
                     if (i, j) in graph.vert_list:
                          for dx in [-1, 0, 1]:
60
                              for dy in [-1, 0, 1]:
61
62
                                  x, y = i + dx, j + dy
                                  if (dx != 0 \text{ or } dy != 0) and 0 <= x < m \text{ and } 0 <=
63
    y < n and (x, y) in graph.vert_list:
64
                                      graph.add\_edge((i, j), (x, y))
65
66
             max_area = 0
             visited = {key: False for key in graph.vert_list.keys()}
67
68
             for i in range(m):
69
                 for j in range(n):
                     if (i, j) in graph.vert_list and not visited[(i, j)]:
70
71
                          vertex = graph.vert_list[(i, j)]
                          area = dfs(graph, vertex, visited)
72
73
                          max\_area = max(max\_area, area)
74
             print(max_area)
75
76
77
    q2()
```

```
אטיגי. Accepted
```

```
      源代码
      #: 44668611

      class Vertex:
      题目: 18160

      def __init__(self, key):
      提交人: 2100017777

      self.id = key
      内存: 4704kB

      self.connected_to = {}
      时间: 197ms

      def add_neighbor(self, nbr, weight=0):
      提交时间: 2024-04-15 21:57:03
```

基本信息

sy383: 最大权值连通块

https://sunnywhy.com/sfbj/10/3/383

思路:约20min.基于第二题改动。

```
1
    class Vertex:
 2
        def __init__(self, key, weight=0):
 3
            self.id = key
            self.connected_to = {}
 4
 5
            self.weight = weight
 6
 7
        def add_neighbor(self, nbr, weight=0):
 8
            self.connected_to[nbr] = weight
9
10
    class Graph:
11
        def __init__(self):
12
            self.vert_list = {}
13
            self.num_vertices = 0
14
15
        def add_vertex(self, key, weight=0):
16
            self.vert_list[key] = Vertex(key, weight)
17
            self.num_vertices += 1
18
19
20
        def __contains__(self, n):
21
            return n in self.vert_list
22
        def add_edge(self, f, t, weight=0):
23
24
            if f not in self.vert_list:
                 self.add_vertex(f)
25
            if t not in self.vert_list:
26
                 self.add_vertex(t)
27
28
            self.vert_list[f].add_neighbor(self.vert_list[t], weight)
29
        def __iter__(self):
30
            return iter(self.vert_list.values())
31
32
33
    def q3():
34
35
```

```
36
        def dfs(graph, vertex, visited):
37
            visited[vertex.id] = True
38
            weight = vertex.weight
39
            for neighbor in vertex.connected_to:
                if not visited[neighbor.id]:
40
                    weight += dfs(graph, neighbor, visited)
41
42
            return weight
43
44
        m, n = map(int, input().split())
45
        weights = list(map(int, input().split()))
        graph = Graph()
46
47
        for i, weight in enumerate(weights):
48
            graph.add_vertex(i, weight)
49
        for _ in range(n):
            a, b = map(int, input().split())
50
51
            graph.add_edge(a, b)
52
            graph.add_edge(b, a)
53
        max\_weight = 0
54
55
        visited = {key: False for key in graph.vert_list.keys()}
56
        for vertex in graph:
57
            if not visited[vertex.id]:
                max_weight = max(max_weight, dfs(graph, vertex, visited))
58
59
60
        print(max_weight)
61
62
    q3()
63
```

```
代码书写
                                              Python
      class Vertex:
  1
          def init (self, key, weight=0):
  3
              self.id = key
  4
              self.connected to = {}
  5
              self.weight = weight
  6
  7
          def add neighbor(self, nbr, weight=0):
  8
              self.connected to[nbr] = weight
  9
测试输入
         提交结果
                  历史提交
 完美通过
                                               查看题解
 100% 数据通过测试
 运行时长: 0 ms
```

03441: 4 Values whose Sum is 0

data structure/binary search, http://cs101.openjudge.cn/practice/03441

思路:约20min.

```
1
    def q4():
 2
        n = int(input())
3
        A, B, C, D = [], [], []
        for _ in range(n):
4
5
            a, b, c, d = map(int, input().split())
 6
            A.append(a)
 7
            B.append(b)
8
            C.append(c)
9
            D.append(d)
10
11
        count = 0
12
        AB\_sums = \{\}
        for a in A:
13
14
            for b in B:
15
                 AB\_sum = a + b
                 if AB_sum in AB_sums:
16
17
                     AB\_sums[AB\_sum] += 1
18
                 else:
```

```
19
                     AB\_sums[AB\_sum] = 1
20
21
        for c in C:
22
             for d in D:
23
                 CD_sum = c + d
24
                 if -CD_sum in AB_sums:
25
                     count += AB_sums[-CD_sum]
26
27
        print(count)
28
29
30
   q4()
```

状态: Accepted

```
      im代码
      #: 44669580

      def q4():
      题目: 03441

      n = int(input())
      提交人: 2100017777

      A, B, C, D = [], [], [], []
      内存: 171608kB

      for _ in range(n):
      时间: 3247ms

      a, b, c, d = map(int, input().split())
      语言: Python3

      A.append(a)
      提交时间: 2024-04-15 23:30:44
```

基本信息

04089: 电话号码

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

Trie 数据结构可能需要自学下。

思路:约50分钟。

```
class Trie:
 2
        def __init__(self):
 3
            self.children = {}
 4
            self.is_end = False
 5
 6
7
    def q5():
8
9
        def insert(t, word):
10
            nonlocal result
11
            if t.is_end:
12
                 result = False
13
                 return
            if not word:
14
15
                 if t.children:
16
                     result = False
17
                     return
```

```
18
                 t.is_end = True
19
            else:
20
                 letter = word[0]
21
                 if letter not in t.children:
                     t.children[letter] = Trie()
22
                 insert(t.children[letter], word[1:])
23
24
25
        for _ in range(int(input())):
26
            t = Trie()
27
            result = True
28
            for i in range(int(input())):
29
                word = input()
30
                 if not result:
31
                     continue
32
                 insert(t, word)
33
            if result:
34
                 print('YES')
35
            else:
                 print('NO')
36
37
38
39
    q5()
```

状态: Accepted

```
      源代码
      #: 44670025

      class Trie:
      题目: 04089

      def __init__(self):
      提交人: 2100017777

      self.children = {}
      内存: 24184kB

      self.is_end = False
      时间: 367ms

      语言: Python3

      提交时间: 2024-04-16 01:20:51
```

基本信息

04082: 树的镜面映射

http://cs101.openjudge.cn/practice/04082/

思路:约40min.

```
1
   class BTree:
       def __init__(self, node, left=None, right=None):
2
3
           self.node = node
           self.left = left
4
5
           self.right = right
6
           self.is_leaf = True
7
8
9
   def q6():
```

```
10
        input()
        raw_nodes = input().split()
11
12
13
        def build():
14
            nonlocal raw_nodes
15
            root = raw_nodes[0]
            t = BTree(root[0])
16
            # wait
17
18
            raw_nodes = raw_nodes[1:]
19
            if root[1] == '0':
                 t.is_leaf = False
20
                 t.left = build()
21
                 t.right = build()
22
23
            return t
24
25
        root_tree = build()
26
        current_level = [root_tree]
27
        next_level = []
        cl_result = []
28
        final_result = []
29
30
        while current_level:
            for t in current_level:
31
32
                 c_n = t
33
                 while True:
34
                     if c_n.node != '$':
35
                         cl_result.append(c_n.node)
36
                     if c_n.right:
37
                         next_level.append(c_n.left)
38
                         c_n = c_n.right
39
                     else: break
40
41
            final_result.extend(cl_result[::-1])
42
            current_level = next_level[:]
            next_level = []
43
44
            cl_result = []
        print(' '.join(final_result))
45
46
47
48
    q6()
```

状态: Accepted

```
#: 44669887

class BTree:
    def __init__(self, node, left=None, right=None):
    self.node = node
    self.left = left
    self.right = right
    self.is_leaf = True

#: 44669887

    题目: 04082

    提交人: 2100017777

    内存: 3680kB

    时间: 26ms
    语言: Python3

    提交时间: 2024-04-16 00:28:30
```

基本信息

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。

第五题的输出需要注意,按照题意,尽管有些NO的情况可以提早判断,依然要等到一组数据全部输入完后再输出NO or YES.