

数据结构与算法 - 综合训练 2

用图解决问题

吴天阳 2204210460 强基数学 002

1 实验目的

给出电影的相关数据, 包含电影名称与参与该电影拍摄的演员. 每行的数据格式为 电影名称/演员 1/演员 2/.../..., 电影名与演员之间均使用左斜杠进行分隔. 对于任何一个演员, 给出该演员到 Kevin Bacon 之间联系所用的 “Bacon Number”, 定义如下:

1. Kevin Bacon 的 Bacon Number 值为 0;
2. 和 Kevin Bacon 在一个电影里出现的所有演员的 Bacon Number 值为 1;
3. 任何演员的 Bacon Number 值为与该演员在同一个电影里的 Bacon Number 值最小的那个演员的 Bacon Number 值加 1.

要求对于给定的某个演员, 求出该演员到演员 Kevin Bacon 的 Bacon Number 并给出计算该数字的依据, 也就是通过哪些电影与 Kevin Bacon 获得联系.

任务 1: 建立为实现该游戏建立的抽象描述结构, 包括图中顶点的意义以及存储的信息、边的意义以及存储的信息. 并给出图的逻辑示意图.

任务 2: 在任务 1 的基础上, 并结合教材中图的抽象数据类型的定义, 设计并实现一个为该游戏而使用的具体的 Graph Class.

任务 3: 根据文件 Simple.txt 构建图, 根据输入的演员名, 给出演员的 Bacon Number, 与其计算依据. (总演员数目为 72 人, 总边数目为 192 条)

任务 4: 根据文件 Complex.txt 构建图, 根据输入的演员名, 给出演员的 Bacon Number, 与其计算依据. (总演员数目为 348567 人, 总边数目为 18005642 条)

任务 5: 在你的日常学习生活中, 寻找一个可以用图解决的问题原型, 描述该问题原型, 并陈述如何将该问题原型抽象成图的表示.

2 实验原理

2.1 任务 1,2

将每个演员视为图中的节点, 与某电影相关的演员, 对应的节点两两之间建立连边.

- 每个节点存储演员的名称, 并用邻接表储存所有出边的指针.
- 每个边均为有向边, 存储对应电影的名称, 并存储该边的头节点与尾节点.

对于每个电影, 假设有 n 个相关的演员, 则一共要添加 $2\binom{n}{2} = n(n-1)$, 首先两两演员之间都要建立双向边联系, 并且由于使用单向边存储, 所以需要建立两个单向边用于表示双向边.

2.2 任务 3,4

使用邻接表存储图结构, 可以对 `Complex.txt` 中图结构进行存储, 只是用时较长. 将读入数据存储为图结构后, 由于每条边的权重均为 1, 可以使用广度优先搜索 (BFS) 查找从 Kevin Bacon 对应节点出发到其他所有节点的最短路径, 该路径长度即为 Bacon Number, 并记录每个节点到达最短路的对应父节点的边, 通过该边可以找到对应的电影名称和父节点, 进而迭代即可复现路径.

2.3 任务 5

一个日常生活中图论的例子是用于研究交通网络中的路线和交通流量.

可将该问题抽象为如下图中的表示: 其中节点表示道路交叉口或地铁站, 边表示道路或地铁线路. 可以使用不同大小边权来表示交通流量或车辆数量. 可以使用图相关的算法来求解最短路径和交通瓶颈.

3 实验步骤与结果分析

3.1 实验 1,2

下面代码实现了节点与边的类, 并使用 `add_edge` 进行边的添加.

```

1 class Graph:
2     def __init__(self):
3         self.total_edge = 0 # 记录总边数
4
5     class Node: # 存储图中节点, 每个演员对应一个节点
6         def __init__(self, name, id):
7             self.id = id
8             self.name = name
9             self.next = [] # 用邻接表存储后继边
10
11     class Edge(): # 存储图中的边, 每个边对应一个电影
12         def __init__(self, movie, previous_node, next_node):
13             self.movie = movie
14             self.previous_node = previous_node
15             self.next_node = next_node # 存储后继节点
16
17     def add_edge(self, node1, node2, movie): # 加入从 node1 到 node2 的单向
18         ↪ 边
19         edge = self.Edge(movie, node1, node2)
20         node1.next.append(edge)
21         self.total_edge += 1

```

以 Apollo 13 (1995)/Bill Paxton/Tom Hanks/Kevin Bacon 为例, 构建关系图如下

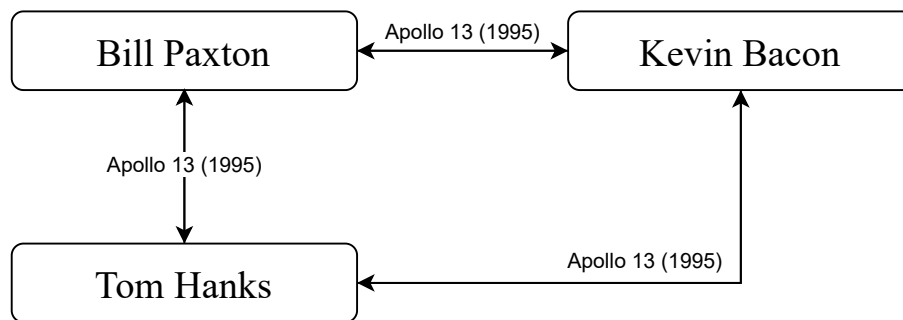


图 1: 逻辑示意图

3.2 任务 3,4

3.2.1 统计与 Kevin Bacon 相关的演员数目

此部分为自己加的额外功能，并非原题中要求。使用并查集判断一共有多少演员与 Kevin Bacon 相关，得出结果 Sample.txt 中有 70 个相关的（总共 72 个），Complex.txt 中有 323278 个相关的（总共 348567 个）。并查集类实现如下

```

1 class Union: # 并查集
2     def __init__(self):
3         self.name_id = 0
4         self.father = []
5
6     def add_id(self): # 加入新 id
7         self.father.append(self.name_id)
8         self.name_id += 1
9         return self.name_id - 1
10
11    def get_father(self, p): # 查询父节点
12        if self.father[p] == p: return p
13        self.father[p] = self.get_father(self.father[p])
14        return self.father[p]
15
16    def join(self, a, b): # 将 a 与 b 的集合合并
17        self.father[self.get_father(a)] = self.get_father(b)

```

3.2.2 核心代码

然后使主求解代码，用于文件读取和广度优先搜索最短距离，广搜从 Kevin Bacon 开始进行。

```

1 class Solver():
2     def __init__(self, fname, kevin_name):
3         self.graph = Graph()
4         self.fname = fname
5         self.kevin_name = kevin_name
6         self.name2node = {} # 字典存储演员对应的节点
7
8     def read_data(self):

```

```

9         union = Union()
10        with open(self.fname, 'r', encoding='utf-8') as file: # Bacon,
11            ↪ Kevin
12            while True:
13                string = file.readline()
14                if not string: break
15
16                items = string.strip()
17                movie = items[0:items.find('')+1] # 电影名称
18                actors = items[items.find('')+2:].split('/') # 演员名称
19
20                for name in actors: # 创建未见过的节点
21                    if name not in self.name2node.keys():
22                        name_id = union.add_id()
23                        self.name2node[name] = Graph.Node(name, name_id) #
24                            ↪ 创建新的节点
25                        union.join(self.name2node[actors[0]].id,
26                            ↪ self.name2node[name].id)
27
28                for i, name1 in enumerate(actors): # 创建边
29                    node1 = self.name2node[name1]
30                    for name2 in actors[i+1:]:
31                        node2 = self.name2node[name2]
32                        self.graph.add_edge(node1, node2, movie)
33                        self.graph.add_edge(node2, node1, movie)
34
35                self.kevin_node = self.name2node[self.kevin_name]
36                self.kevin_id = self.kevin_node.id
37                relation_node = 0
38                for i in range(union.name_id): # 计算与 Kevin Bacon 相关的演员数目
39                    if union.get_father(i) == union.get_father(self.kevin_id):
40                        relation_node += 1
41
42                print(" 总演员数目:", len(self.name2node))
43                print(" 总边数目:", self.graph.total_edge)
44                print(f" 和{self.kevin_name}相关的演员总数:", relation_node)
45
46        def bfs(self): # 广度优先搜索
47            from queue import Queue
48            self.father_edge = {} # 记录连接到父节点的边
49            visited = {self.kevin_node} # 判断是否访问过该节点
50            self.distance = {self.kevin_name: 0} # 记录最短距离
51            q = Queue()
52            q.put(self.kevin_node) # 加入 Kevin 节点
53            while not q.empty():
54                u = q.get()
55                for e in u.next:
56                    v = e.next_node # 访问新的节点
57                    if v in visited: continue
58                    visited.add(v)

```

```

56         self.distance[v.name] = self.distance[u.name] + 1 # 更新距
           ↳ 离
57         self.father_edge[v] = e # 记录连接父节点的边
58         q.put(v)
59
60     def show_path(self, name): # 回溯打印路径
61         if name not in self.distance.keys(): # 若无法到达该节点
62             print(f"\nCan't find path from {name} to {self.kevin_name}.")
63             return
64         print(f"\nPath from {name} to {self.kevin_name}:")
65         node = self.name2node[name]
66         while node.name != self.kevin_name: # 利用连接父节点的边, 复现路径
67             father_edge = self.father_edge[node]
68             father_node = father_edge.previous_node
69             print(f"{node.name} was in {father_edge.movie} with
           ↳ {father_node.name}")
70             node = father_node
71         print(f"{name}'s Bacon number is {self.distance[name]}")

```

最后是主程序代码:

```

1  if __name__ == '__main__':
2      # solver = Solver("Simple.txt", "Kevin Bacon")
3      solver = Solver("Complex.txt", "Bacon, Kevin")
4      start_time = time.time()
5      solver.read_data()
6      solver.bfs()
7      print(" 预处理用时:", time.time() - start_time, "s")
8      print(" 距离 Kevin 的最大 Bacon 距离:", max(solver.distance.values()))
9      while True:
10         command = input("Actor's name (or All for everyone or Show Bacon
           ↳ distance bigger than NUMBER)?\n> ")
11         if command == 'All':
12             for name in solver.name2node.keys():
13                 solver.show_path(name)
14         elif 'Show Bacon distance bigger than' in command: # 新命令, 可显示
           ↳ Bacon 距离 >= 某个值的全部节点
15             num = int(command.split()[-1])
16             for name, distance in solver.distance.items():
17                 if distance >= num:
18                     solver.show_path(name)
19         else:
20             solver.show_path(command)

```

3.2.3 任务 3 执行结果

对于文件 Sample.txt, 执行效果如下 (>右侧为用户输入的文本):

```

1 总演员数目: 72
2 总边数目: 192
3 和 Kevin Bacon 相关的演员总数: 70
4 预处理用时: 0.012998819351196289

```

```
5 距离 Kevin 的最大 Bacon 距离: 5
6 Actor's name (or All for everyone or Show Bacon distance bigger than
  ↳ NUMBER)?
7 > Brad Pitt
8
9 Path from Brad Pitt to Kevin Bacon:
10 Brad Pitt was in Ocean's Eleven (2001) with Julia Roberts
11 Julia Roberts was in Flatliners (1990) with Kevin Bacon
12 Brad Pitt's Bacon number is 2
13
14 Actor's name (or All for everyone or Show Bacon distance bigger than
  ↳ NUMBER)?
15 > Show Bacon distance bigger than 5
16
17 Path from P. Biryukov to Kevin Bacon:
18 P. Biryukov was in Pikovaya dama (1910) with Aleksandr Gromov
19 Aleksandr Gromov was in Tikhij Don (1930) with Yelena Maksimova
20 Yelena Maksimova was in Bezottsovshchina (1976) with Lev Prygunov
21 Lev Prygunov was in Saint, The (1997) with Elisabeth Shue
22 Elisabeth Shue was in Hollow Man (2000) with Kevin Bacon
23 P. Biryukov's Bacon number is 5
24
25
26 Path from Yelena Chaika to Kevin Bacon:
27 Yelena Chaika was in Ostrov zabenyia (1917) with Viktor Tourjansky
28 Viktor Tourjansky was in Zagrobnaya skitalitsa (1915) with Olga Baclanova
29 Olga Baclanova was in Freaks (1932) with Angelo Rossitto
30 Angelo Rossitto was in Dark, The (1979) with William Devane
31 William Devane was in Hollow Man (2000) with Kevin Bacon
32 Yelena Chaika's Bacon number is 5
33
34
35 Path from Zoya Barantsevich to Kevin Bacon:
36 Zoya Barantsevich was in Slesar i kantzler (1923) with Nikolai Panov
37 Nikolai Panov was in Zhenshchina s kinzhalom (1916) with Zoia Karabanova
38 Zoia Karabanova was in Song to Remember, A (1945) with William Challee
39 William Challee was in Irish Whiskey Rebellion (1972) with William Devane
40 William Devane was in Hollow Man (2000) with Kevin Bacon
41 Zoya Barantsevich's Bacon number is 5
42
43
44 Path from Christel Holch to Kevin Bacon:
45 Christel Holch was in Hvide Slavehandel, Den (1910/I) with Aage Schmidt
46 Aage Schmidt was in Begyndte ombord, Det (1937) with Valso Holm
47 Valso Holm was in Spion 503 (1958) with Max von Sydow
48 Max von Sydow was in Judge Dredd (1995) with Diane Lane
49 Diane Lane was in My Dog Skip (2000) with Kevin Bacon
50 Christel Holch's Bacon number is 5
```

3.2.4 任务4 执行结果

对于文件 Complex.txt, 执行效果如下 (>右侧为用户输入的文本):

```

1 总演员数目: 348567
2 总边数目: 18005642
3 和 Bacon, Kevin 相关的演员总数: 323278
4 预处理用时: 39.396482944488525 s
5 距离 Kevin 的最大 Bacon 距离: 9
6 Actor's name (or All for everyone or Show Bacon distance bigger than
  ↳ NUMBER)?
7 > Decleir, Jenne
8
9 Path from Decleir, Jenne to Bacon, Kevin:
10 Decleir, Jenne was in Verlossing, De (2001) with Ammelrooy, Willeke van
11 Ammelrooy, Willeke van was in Lake House, The (2006) with Bullock, Sandra
12 Bullock, Sandra was in Loverboy (2005) with Bacon, Kevin
13 Decleir, Jenne's Bacon number is 3
14
15 Actor's name (or All for everyone or Show Bacon distance bigger than
  ↳ NUMBER)?
16 > Show Bacon distance bigger than 9
17
18 Path from Schieferdecker, Daniel to Bacon, Kevin:
19 Schieferdecker, Daniel was in Soundcheck (2001) with Urich, Christian
20 Urich, Christian was in Erste Nacht, Die (2003) with Havemann, Lars
21 Havemann, Lars was in Sandzeit (2005) with Schilling, Lea
22 Schilling, Lea was in Was denkt man, wenn... (2004) with Landsiedel, Timo
23 Landsiedel, Timo was in Wagnisse (2006) with Schumacher, Christian (II)
24 Schumacher, Christian (II) was in Bißchen Mord muß sein, Ein (2000) with
  ↳ Alfieri, Vittorio (I)
25 Alfieri, Vittorio (I) was in Beyond the Sea (2004) with Goodman, John (I)
26 Goodman, John (I) was in Clifford's Really Big Movie (2004) with
  ↳ Valderrama, Wilmer
27 Valderrama, Wilmer was in Beauty Shop (2005) with Bacon, Kevin
28 Schieferdecker, Daniel's Bacon number is 9
29
30
31 Path from Heintze, Christoph to Bacon, Kevin:
32 Heintze, Christoph was in Soundcheck (2001) with Urich, Christian
33 Urich, Christian was in Erste Nacht, Die (2003) with Havemann, Lars
34 Havemann, Lars was in Sandzeit (2005) with Schilling, Lea
35 Schilling, Lea was in Was denkt man, wenn... (2004) with Landsiedel, Timo
36 Landsiedel, Timo was in Wagnisse (2006) with Schumacher, Christian (II)
37 Schumacher, Christian (II) was in Bißchen Mord muß sein, Ein (2000) with
  ↳ Alfieri, Vittorio (I)
38 Alfieri, Vittorio (I) was in Beyond the Sea (2004) with Goodman, John (I)
39 Goodman, John (I) was in Clifford's Really Big Movie (2004) with
  ↳ Valderrama, Wilmer
40 Valderrama, Wilmer was in Beauty Shop (2005) with Bacon, Kevin
41 Heintze, Christoph's Bacon number is 9
42

```

```

43
44 Path from De Silva, Rahel to Bacon, Kevin:
45 De Silva, Rahel was in Soundcheck (2001) with Urich, Christian
46 Urich, Christian was in Erste Nacht, Die (2003) with Havemann, Lars
47 Havemann, Lars was in Sandzeit (2005) with Schilling, Lea
48 Schilling, Lea was in Was denkt man, wenn... (2004) with Landsiedel, Timo
49 Landsiedel, Timo was in Wagnisse (2006) with Schumacher, Christian (II)
50 Schumacher, Christian (II) was in Bißchen Mord muß sein, Ein (2000) with
   ↳ Alfieri, Vittorio (I)
51 Alfieri, Vittorio (I) was in Beyond the Sea (2004) with Goodman, John (I)
52 Goodman, John (I) was in Clifford's Really Big Movie (2004) with
   ↳ Valderrama, Wilmer
53 Valderrama, Wilmer was in Beauty Shop (2005) with Bacon, Kevin
54 De Silva, Rahel's Bacon number is 9
55
56
57 Path from Wilken, Sonja (II) to Bacon, Kevin:
58 Wilken, Sonja (II) was in Soundcheck (2001) with Urich, Christian
59 Urich, Christian was in Erste Nacht, Die (2003) with Havemann, Lars
60 Havemann, Lars was in Sandzeit (2005) with Schilling, Lea
61 Schilling, Lea was in Was denkt man, wenn... (2004) with Landsiedel, Timo
62 Landsiedel, Timo was in Wagnisse (2006) with Schumacher, Christian (II)
63 Schumacher, Christian (II) was in Bißchen Mord muß sein, Ein (2000) with
   ↳ Alfieri, Vittorio (I)
64 Alfieri, Vittorio (I) was in Beyond the Sea (2004) with Goodman, John (I)
65 Goodman, John (I) was in Clifford's Really Big Movie (2004) with
   ↳ Valderrama, Wilmer
66 Valderrama, Wilmer was in Beauty Shop (2005) with Bacon, Kevin
67 Wilken, Sonja (II)'s Bacon number is 9

```

4 结论与讨论

通过本次实验，学会了如何用邻接表存储图结构，将人际关系网络图转化为图论问题，进而求解两两人物之间的关系路径，提升将问题进行转化的能力。学会如何用 Python 求解单源最短路径，使用不同的类完成固定的任务，学会如何将整体任务划分为较小的部分分块解决。