

Assignment #A: Graph starts

Updated 1830 GMT+8 Apr 22, 2025

2025 spring, Compiled by 颜鼎堃 工学院

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

M19943:图的拉普拉斯矩阵

OOP, implementation, <http://cs101.openjudge.cn/practice/19943/>

要求创建Graph, Vertex两个类，建图实现。

思路：

- 想起来线性代数课上老师讲过这个

代码：

```
1 class Vertex:
2     def __init__(self, n, D=[]):
3         self.D = D
4         for i in range(n):
5             for j in range(n):
6                 D[-1].append(0)
7             D.append([])
8 class Graph:
9     def __init__(self, n, A=[]):
10        self.A = A
11        for i in range(n):
12            for j in range(n):
13                A[-1].append(0)
14            A.append([])
15
16 n, m = map(int, input().split())
17 v = [Vertex(n)] * n
18 G = Graph(n)
19 ans = [[0 for j in range(n)] for i in range(n)]
20 for i in range(m):
21     v1, v2 = map(int, input().split())
22     v[v1].D[v1][v1] += 1
23     v[v2].D[v2][v2] += 1
24     G.A[v1][v2] = G.A[v2][v1] = 1
25 for i in range(n):
26     for j in range(n):
27         ans[i][j] = v[0].D[i][j] - G.A[i][j]
```

Python

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

 **CS101 / 题库 (包括计概、数算题目)**

题目 排名 状态 提问

#48998454提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class Vertex:
    def __init__(self, n, D=[]):
        self.D = D
        for i in range(n):
            for j in range(n):
                D[-1].append(0)
            D.append([])

class Graph:
    def __init__(self, n, A=[]):
        self.A = A
        for i in range(n):
            for j in range(n):
                A[-1].append(0)
            A.append([])

n, m = map(int, input().split())
v = [Vertex(n)] * n
G = Graph(n)
ans = [[0 for j in range(n)] for i in range(n)]
for i in range(m):
    v1, v2 = map(int, input().split())
    v[v1].D[v1][v1] += 1
    v[v2].D[v2][v2] += 1
    G.A[v1][v2] = G.A[v2][v1] = 1
for i in range(n):
    for j in range(n):
        ans[i][j] = v[i].D[i][j] - G.A[i][j]
    print(*ans[i])
```

基本信息

#:

48998454

题目:

19943

提交人:

颜鼎堃(24n2400011125)

内存:

3672kB

时间:

19ms

语言:

Python3

提交时间:

2025-04-24 11:11:49

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

LC78.子集

backtracking, <https://leetcode.cn/problems/subsets/>

思路:

- 其实我一直没理解回溯是什么意思, 反正大概应该就是递归吧

代码:

```
1  from typing import *
2  class Solution:
3      def subsets(self, nums: List[int]) → List[List[int]]:
4          self.ans = []
5          def recur(nums, depth, pref):
6              if depth == len(nums):
7                  self.ans.append(pref)
8                  return
9              recur(nums, depth + 1, pref)
10             recur(nums, depth + 1, pref + [nums[depth]])
11
12             recur(nums, 0, [])
13             return self.ans
14
15 if __name__ == '__main__':
16     sol = Solution()
17     print(sol.subsets([1, 2, 3]))
```

Python

78. 子集 已解答 ✓

中等 相关标签 相关企业 Aa

给你一个整数数组 `nums`，数组中的元素 **互不相同**。返回该数组所有可能的**子集**（幂集）。

解集 **不能** 包含重复的子集。你可以按 **任意顺序** 返回解集。

示例 1:

输入: `nums = [1,2,3]`
 输出: `[[], [1], [2], [1,2], [3], [1,3], [2,3], [1,2,3]]`

示例 2:

输入: `nums = [0]`
 输出: `[[], [0]]`

提示:

- `1 <= nums.length <= 10`
- `-10 <= nums[i] <= 10`
- `nums` 中的所有元素 **互不相同**

面试中遇到过这道题? 1/5

是 否

通过次数 997.7K | 提交次数 1.2M | 通过率 82.2%

2.5K 2.5K ☆ ② 86 人在线

代码 Python3 智能模式

```

1 class Solution:
2     def subsets(self, nums: List[int]) -> List[List[int]]:
3         self.ans = []
4         def recur(nums, depth, pref):
5             if depth == len(nums):
6                 self.ans.append(pref)
7                 return
8             recur(nums, depth + 1, pref)
9             recur(nums, depth + 1, pref + [nums[depth]])
10
11         recur(nums, 0, [])
12         return self.ans
13

```

行 1, 列 1 | 已从本地恢复 升级云端代码存储 运行 提交

测试用例 测试结果

Case 1 Case 2 +

nums =

[1,2,3]

</> Source ②

LC17.电话号码的字母组合

hash table, backtracking, <https://leetcode.cn/problems/letter-combinations-of-a-phone-number/>

思路:

- 跟子集差不多

代码:

```

1 from typing import *
2 class Solution:
3     def letterCombinations(self, digits: str) -> List[str]:
4         NUM_TO_LETTER = {"2": "abc", "3": "def", "4": "ghi", "5": "jkl", "6": "mno", "7": "pqrs", "8": "tuv",
5         "9": "wxyz"}
6         self.ans = []
7         def recur(digits, depth, prev):
8             if depth >= len(digits):
9                 self.ans.append(prev)
10                return
11            for n in NUM_TO_LETTER[digits[depth]]:
12                recur(digits, depth + 1, prev + n)
13            recur(digits, 0, "")
14            if not self.ans[0]:
15                self.ans.pop()
16            return self.ans
17
18 if __name__ == "__main__":
19     sol = Solution()
20     print(sol.letterCombinations(""))

```

Python

已解答

中等 相关标签 相关企业 Aa

给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。



输入: digits = "23"

输出: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

示例 2:

输入: `digits = ""`

输出: []

示例 3:

输入: digits = "2"

输出: ["a", "b", "c"]

提示：

3.1K 3.8K

● 109 人在线

</> 代码

Python3 智能模式

```

1 class Solution:
2     def letterCombinations(self, digits: str) -> List[str]:
3         NUM_TO_LETTER = {"2": "abc", "3": "def", "4": "ghi", "5": "jkl", "6": "mno", "7": "pqrs",
4 "8": "tuv", "9": "wxyz"}
5         self.ans = []
6         def recur(digits, depth, prev):
7             if depth >= len(digits):
8                 self.ans.append(prev)
9                 return
10            for n in NUM_TO_LETTER[digits[depth]]:
11                recur(digits, depth + 1, prev + n)
12        recur(digits, 0, "")
13        if not self.ans[0]:
14            self.ans.pop()
15        return self.ans

```

行 1, 列 1 | 已存儲

运行

提交

☑ 测试用例 | > 测试结果

Case 1

Case 2

Case

+

digits =

"23"

Source ?

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

- 理解了前缀树会发现没那么难
- 最开始写的时候没考虑到可能会有相同的电话号码，我还专门把相同的电话号码设成了不矛盾
- 也就是下面第14行写的是 `if p and "end" not in p:`
- 确实不太合理，不同的人怎么会用同一个电话号码呢
- 最后问了小北explore，回答是挺胡扯八道的，但代码是对的
- ai还是厉害，ai快取代我吧

```

1 class Trie:
2     def __init__(self):
3         self.trie = {}
4     def add(self, num):
5         p = self.trie
6         for n in num:
7             if "end" in p:
8                 return False
9             if n in p:
10                p = p[n]
11            else:
12                p[n] = {}
13                p = p[n]
14        if p:
15            return False
16        p["end"] = True
17        return True
18
19 for i in range(int(input())):
20     t = Trie()

```

Python

```

21     flag = True
22     for i in range(int(input())):
23         flag &= t.add(input())
24     print("YES" if flag else "NO")

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#49014108提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

class Trie:
    def __init__(self):
        self.trie = {}
    def add(self, num):
        p = self.trie
        for n in num:
            if "end" in p:
                return False
            if n in p:
                p = p[n]
            else:
                p[n] = {}
                p = p[n]
        if p:
            return False
        p["end"] = True
        return True

for i in range(int(input())):
    t = Trie()
    flag = True
    for i in range(int(input())):
        flag &= t.add(input())
    print("YES" if flag else "NO")

```

基本信息

#: 49014108

题目: 04089

提交人: 颜鼎堃(24n2400011125)

内存: 16852kB

时间: 187ms

语言: Python3

提交时间: 2025-04-26 09:55:17

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

T28046:词梯

bfs, <http://cs101.openjudge.cn/practice/28046/>

思路:

- 上课听到了做桶的思路，所以整体做起来没那么难

代码:

```

1  from collections import deque, defaultdict
2  def find_barrel(word):
3      for c in range(len(word)):
4          yield word[:c] + " " + word[c+1:]
5  words = [input() for i in range(int(input()))]
6  barrels = defaultdict(list)
7  for word in words:
8      for b in find_barrel(word):
9          barrels[b].append(word)
10 start, end = input().split()
11 queue = deque([(start, [])])
12 inq = {start}
13 while queue:
14     node, route = queue.popleft()
15     if node == end:
16         print(*(route + [node]))
17         exit()
18     for b in find_barrel(node):
19         for w in barrels[b]:
20             if w not in inq and b != node:
21                 inq.add(w)
22                 queue.append((w, route + [node]))

```

Python

```
23
24 print("NO")
```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#49016879提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque, defaultdict
def find_barrel(word):
    for c in range(len(word)):
        yield word[:c] + " " + word[c+1:]
words = [input() for i in range(int(input()))]
barrels = defaultdict(list)
for word in words:
    for b in find_barrel(word):
        barrels[b].append(word)
start, end = input().split()
queue = deque([(start, [])])
inq = {start}
while queue:
    node, route = queue.popleft()
    if node == end:
        print(*route + [node])
        exit()
    for b in find_barrel(node):
        for w in barrels[b]:
            if w not in inq and b != node:
                inq.add(w)
                queue.append((w, route + [node]))

print("NO")
```

print("NO")

基本信息

#: 49016879

题目: 28046

提交人: 颜鼎堃(24n2400011125)

内存: 5796kB

时间: 49ms

语言: Python3

提交时间: 2025-04-26 15:00:43

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

T51.N皇后

backtracking, <https://leetcode.cn/problems/n-queens/>

思路:

- 跟上学期八皇后一个思路

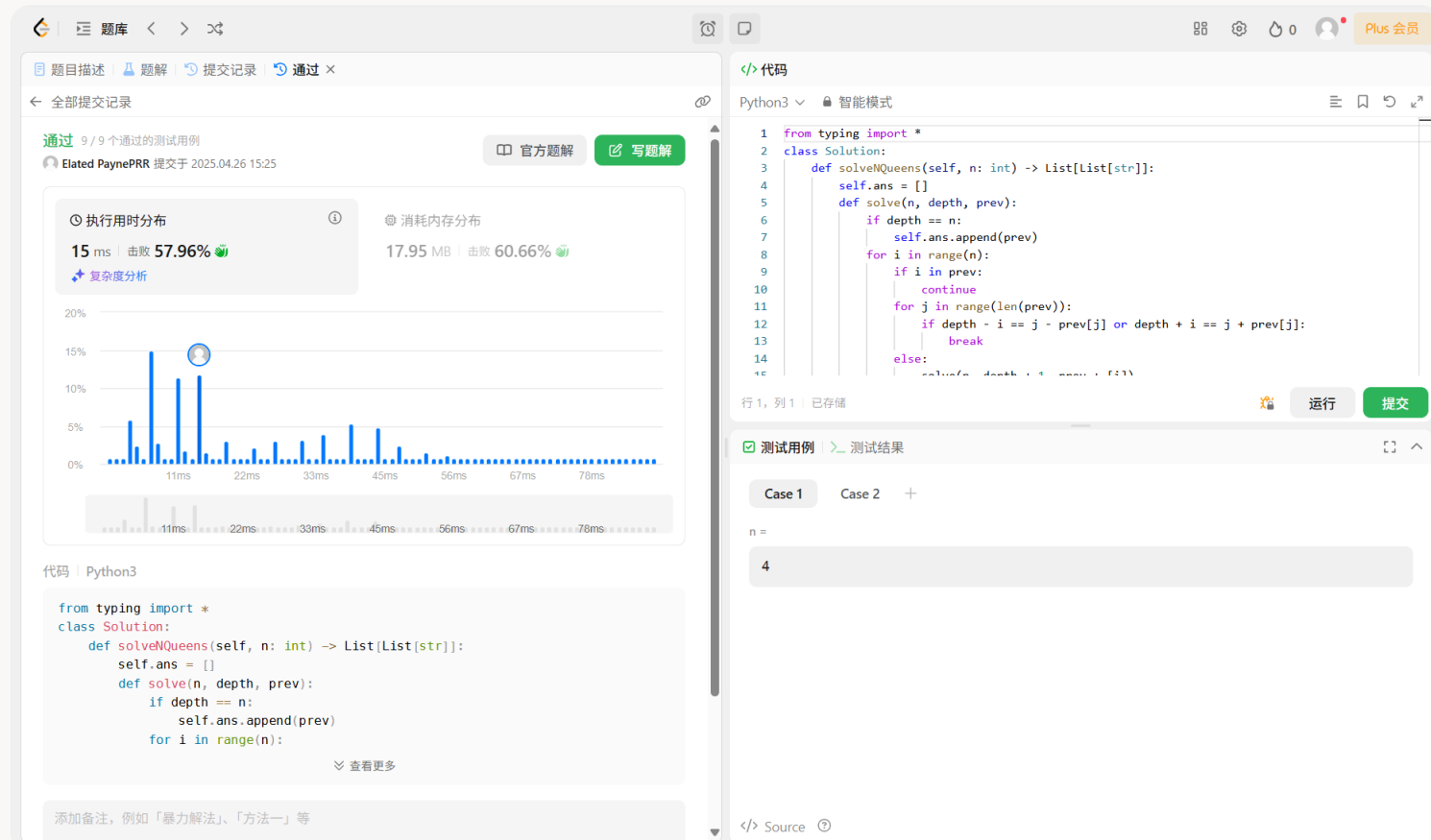
代码:

```
1 from typing import *
2 class Solution:
3     def solveNQueens(self, n: int) -> List[List[str]]:
4         self.ans = []
5         def solve(n, depth, prev):
6             if depth == n:
7                 self.ans.append(prev)
8             for i in range(n):
9                 if i in prev:
10                     continue
11                 for j in range(len(prev)):
12                     if depth - i == j - prev[j] or depth + i == j + prev[j]:
13                         break
14                 else:
15                     solve(n, depth + 1, prev + [i])
16
17         solve(n, 0, [])
18         board = []
19         for a in self.ans:
20             board.append(["." .join(["." if i != a[j] else "Q" for i in range(n)]) for j in range(n)])
21         return board
22
23
24 if __name__ == "__main__":
```

Python

```
25 sol = Solution()
26 print(sol.solveNQueens(4))
27
```

代码运行截图（至少包含有"Accepted"）



2. 学习总结和收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

这次作业因为上课听了点思路，做起来不是很难。如果上课没听到剧透，自己确实很难做出来了