

Assignment #A: Graph starts

Updated 1830 GMT+8 Apr 22, 2025

2025 spring, Compiled by 胡新璞, 工学院

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：不太理解用 Oop 算法的意义（因为直接暴力写比较容易），看了看群里的交流又琢磨了下要求的题解，复习了下 oop 算法，没有再写。代码我就提交我直接写的（）

代码：

```
n,m = map(int,input().split())
bian = []
bian_cnt = [0] * n
for _ in range(m):
    node1,node2 = map(int,input().split())
    bian.append((node1,node2))
    bian_cnt[node1] += 1
    bian_cnt[node2] += 1
matrix = [[0] * n for _ in range(n)]
for i in range(n):
    for j in range(n):
        if i == j:
            matrix[i][j] = bian_cnt[i]
        else:
            if (i,j) in bian or (j,i) in bian:
                matrix[i][j] = -1
for i in range(n):
    print(" ".join(map(str,matrix[i])))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
n,m = map(int,input().split())
bian = []
bian_cnt = [0] * n
for _ in range(m):
    node1,node2 = map(int,input().split())
    bian.append((node1,node2))
    bian_cnt[node1] += 1
    bian_cnt[node2] += 1
matrix = [[0] * n for _ in range(n)]
for i in range(n):
    for j in range(n):
        if i == j:
            matrix[i][j] = bian_cnt[i]
        else:
            if (i,j) in bian or (j,i) in bian:
                matrix[i][j] = -1
for i in range(n):
    print(" ".join(map(str,matrix[i])))
```

基本信息

#: 49032208
题目: 19943
提交人: 2400011037
内存: 3644kB
时间: 21ms
语言: Python3
提交时间: 2025-04-28 22:02:59

LC78.子集

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

思路：迭代那解法有点意思，没想到还能这么写（见的题还是少了），dfs 比较常规。

代码：

class Solution:

```
def subsets(self, nums: List[int]) -> List[List[int]]:
```

```
    ans = []
```

```
    def dfs(num,subset):
```

```
        ans.append(subset)
```

```
        for i in range(num,len(nums)):
```

```
            dfs(i + 1, subset + [nums[i]])
```

```
    dfs(0,[])
```

```
    return ans
```

代码运行截图 （至少包含有"Accepted"）

The screenshot displays the LeetCode submission interface for problem 78. On the left, a status bar indicates '通过' (Accepted) for 10/10 test cases, submitted on 2025.04.28 at 22:12. Below this is a promotional banner for students. The main section shows the Python code for the 'subsets' method using DFS. On the right, the execution results are shown: '执行用时分布' (Execution Time Distribution) is 0 ms with a 100.00% success rate, and '消耗内存分布' (Memory Usage Distribution) is 17.59 MB with a 58.83% success rate. The '测试用例' (Test Cases) section shows 'Case 1' with 'nums = '.

通过 10 / 10 个通过的测试用例
提交于 2025.04.28 22:12

官方题解 写题解

面向在校学生的专享特惠
完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助

执行用时分布
0 ms | 击败 100.00%
复杂度分析

消耗内存分布
17.59 MB | 击败 58.83%

```
1 class Solution:
2     def subsets(self, nums: List[int]) -> List[List[int]]:
3         ans = []
4         def dfs(num,subset):
5             ans.append(subset)
6             for i in range(num,len(nums)):
7                 dfs(i + 1, subset + [nums[i]])
8         dfs(0,[])
9         return ans
```

已存储 行 9.

测试用例

Case 1 Case 2 +

nums =

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：也是比较常规的 dfs，需要注意输入为空，参考示例里给出来了比较友好。

代码：

class Solution:

```
def letterCombinations(self, digits: str) -> List[str]:
    f = {2:"abc",3:"def",4:"ghi",5:"jkl",6:"mno",7:"pqrs",8:"tuv",9:"wxyz"}
    ans = []
    def dfs(tmp,i):
        if i == len(digits):
            ans.append(tmp)
            return
        alphabet = f[int(digits[i])]
        for j in range(len(alphabet)):
            dfs(tmp+alphabet[j],i+1)
    if digits:
        dfs("",0)
    return ans
```

代码运行截图（至少包含有"Accepted"）

The screenshot displays the LeetCode interface for the problem "Letter Combinations of a Phone Number". It includes the following elements:

- Submission Status:** "通过" (Accepted) with 25/25 test cases passed. The submission was made on 2025.04.28 at 22:43.
- Performance Metrics:**
 - Execution Time:** 0 ms, 100.00% beat rate.
 - Memory Usage:** 17.51 MB, 60.28% beat rate.
- Code Editor:** Shows the Python code for the solution, which is a DFS-based approach.
- Test Cases:** A section for testing the solution, with "Case 1" selected. The input "digits =" is visible.

M04089:电话号码

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

思路：（好像也并不一定要用 Trie。看了下题解应该是因为数据还不够多？之后会再看题解学 Trie 的写法。）

代码：

```
t = int(input())
for _ in range(t):
    n = int(input())
    phones = [input() for _ in range(n)]
    phones.sort()
    flag = False
    for i in range(n-1):
        if phones[i+1][:len(phones[i])] == phones[i]:
            flag = True
            break
    print("NO" if flag else "YES")
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
t = int(input())
for _ in range(t):
    n = int(input())
    phones = [input() for _ in range(n)]
    phones.sort()
    flag = False
    for i in range(n-1):
        if phones[i+1][:len(phones[i])] == phones[i]:
            flag = True
            break
    print("NO" if flag else "YES")
```

基本信息

#: 49032440
题目: 04089
提交人: 2400011037
内存: 5368kB
时间: 80ms
语言: Python3
提交时间: 2025-04-28 23:02:20

T28046:词梯

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

思路：

代码：

代码运行截图 （至少包含有"Accepted"）

T51.N 皇后

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

思路：

代码：

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。

这周偷点小懒采用了一些投机的计概学会的写法（）等我期中考完五一一定恶补数算。