Assignment #4: Matrix, 链表 & Backtracking

Updated 2226 GMT+8 Sep 29, 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

E18161: 矩阵运算

matrices, http://cs101.openjudge.cn/pctbook/E18161/

请使用 `@` 矩阵相乘运算符。

思路：

代码：

```
class Matrix:
    def __init__(self, data):
        self.data = data
        self.rows = len(data)
        self.cols = len(data[0]) if self.rows else 0

    def __matmul__(self, other):
```

```
        if self.cols != other.rows:
            raise ValueError
        result = [[0] * other.cols for _ in range(self.rows)]
        for i in range(self.rows):
            for j in range(other.cols):
                for k in range(self.cols):
                    result[i][j] += self.data[i][k] * other.data[k][j]
        return Matrix(result)

    def __add__(self, other):
        if self.rows != other.rows or self.cols != other.cols:
            raise ValueError
        result = [[self.data[i][j] + other.data[i][j] for j in
range(self.cols)] for i in range(self.rows)]
        return Matrix(result)

    def __str__(self):
        return "\n".join(" ".join(map(str, row)) for row in self.data)

def read_matrix():
    r, c = map(int, input().split())
    data = [list(map(int, input().split())) for _ in range(r)]
    return Matrix(data)

A = read_matrix()
B = read_matrix()
C = read_matrix()

try:
    D = A @ B + C
    print(D)
except ValueError:
    print("Error!")
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

基本信息

#: 50275277
题目: E18161
提交人: 25n2400011575
内存: 4412kB
时间: 92ms
语言: Python3
提交时间: 2025-10-09 16:19:49

源代码

```
class Matrix:
    def __init__(self, data):
        self.data = data
        self.rows = len(data)
        self.cols = len(data[0]) if self.rows else 0

    def __matmul__(self, other):
        if self.cols != other.rows:
            raise ValueError
        result = [[0] * other.cols for _ in range(self.rows)]
        for i in range(self.rows):
            for j in range(other.cols):
                for k in range(self.cols):
                    result[i][j] += self.data[i][k] * other.data[k][j]
        return Matrix(result)

    def __add__(self, other):
        if self.rows != other.rows or self.cols != other.cols:
            raise ValueError
        result = [[self.data[i][j] + other.data[i][j] for j in range(sel
        return Matrix(result)

    def __str__(self):
        return "\n".join(" ".join(map(str, row)) for row in self.data)

def read_matrix():
    r, c = map(int, input().split())
    data = [list(map(int, input().split())) for _ in range(r)]
    return Matrix(data)

A = read_matrix()
B = read_matrix()
C = read_matrix()

try:
    D = A @ B + C
    print(D)
except ValueError:
    print("Error!")
```

用时：10min左右

E19942: 二维矩阵上的卷积运算

matrices, http://cs101.openjudge.cn/pctbook/E19942/

思路：

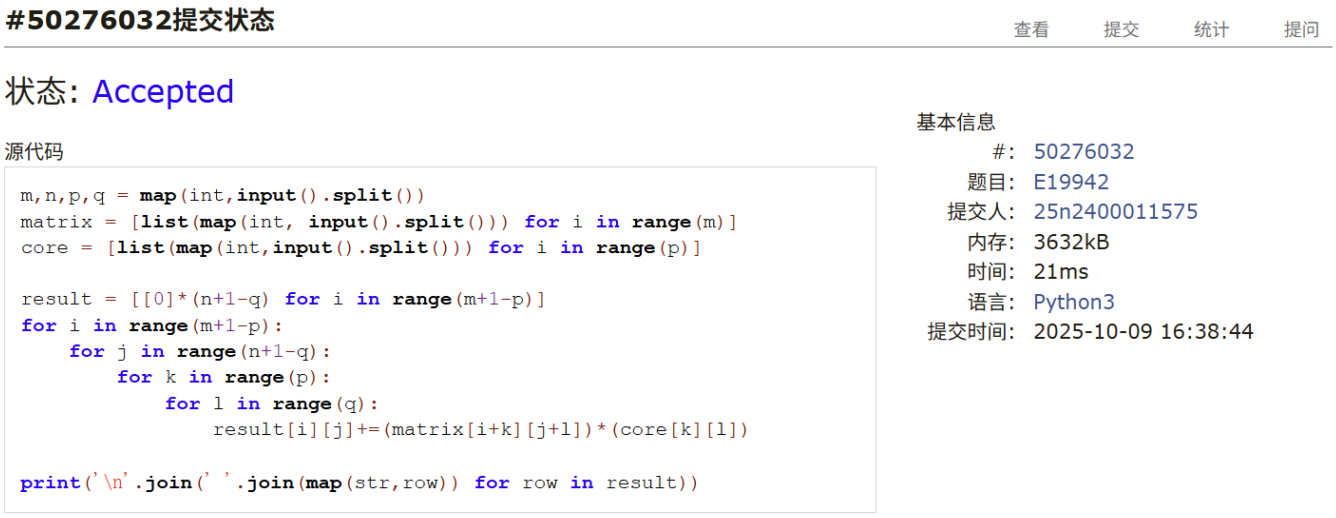
关键是找准最终结果作为一个矩阵，i行j列的元素由什么得来。确定遍历对象，踏实计算，适当举例。

代码：

```
m,n,p,q = map(int,input().split())
matrix = [list(map(int, input().split())) for i in range(m)]
```

```
core = [list(map(int,input().split())) for i in range(p)]

result = [[0]*(n+1-q) for i in range(m+1-p)]
for i in range(m+1-p):
    for j in range(n+1-q):
        for k in range(p):
            for l in range(q):
                result[i][j]+=(matrix[i+k][j+l])*(core[k][l])

print('\n'.join(' '.join(map(str,row)) for row in result))
```

代码运行截图 （至少包含有"Accepted"）



#50276032提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
m,n,p,q = map(int,input().split())
matrix = [list(map(int, input().split())) for i in range(m)]
core = [list(map(int,input().split())) for i in range(p)]

result = [[0]*(n+1-q) for i in range(m+1-p)]
for i in range(m+1-p):
    for j in range(n+1-q):
        for k in range(p):
            for l in range(q):
                result[i][j]+=(matrix[i+k][j+l])*(core[k][l])

print('\n'.join(' '.join(map(str,row)) for row in result))
```

基本信息

#: 50276032
题目: E19942
提交人: 25n2400011575
内存: 3632kB
时间: 21ms
语言: Python3
提交时间: 2025-10-09 16:38:44

用时：13min左右

M06640: 倒排索引

data structures, http://cs101.openjudge.cn/pctbook/M06640/

思路：

用字典将word作为key，在value中进行文件索引的存储。利用0表示不存在。利用setdefault方法设置默认值。

代码：

```
n = int(input())
dic = {}
for i in range(n):
    s = input()[1:].split()
    for j in set(s):
        dic.setdefault(j,[]).append(i+1)
```

```
m = int(input())
for i in range(m):
    word = input()
    out = sorted(dic.setdefault(word,[0]))
    out = list(map(str,out))
    print(' '.join(out) if out[0]!='0' else 'NOT FOUND')
```

代码运行截图 （至少包含有"Accepted"）

#50277196提交状态

查看 提交 统计 提

状态: Accepted

源代码

```
n = int(input())
dic = {}
for i in range(n):
    s = input()[1:].split()
    for j in set(s):
        dic.setdefault(j,[]).append(i+1)

m = int(input())
for i in range(m):
    word = input()
    out = sorted(dic.setdefault(word,[0]))
    out = list(map(str,out))
    print(' '.join(out) if out[0]!='0' else 'NOT FOUND')
```

基本信息

#: 50277196
题目: M06640
提交人: 25n2400011575
内存: 6900kB
时间: 64ms
语言: Python3
提交时间: 2025-10-09 17:11:33

用时：30min左右

E160.相交链表

two pinters, https://leetcode.cn/problems/intersection-of-two-linked-lists/

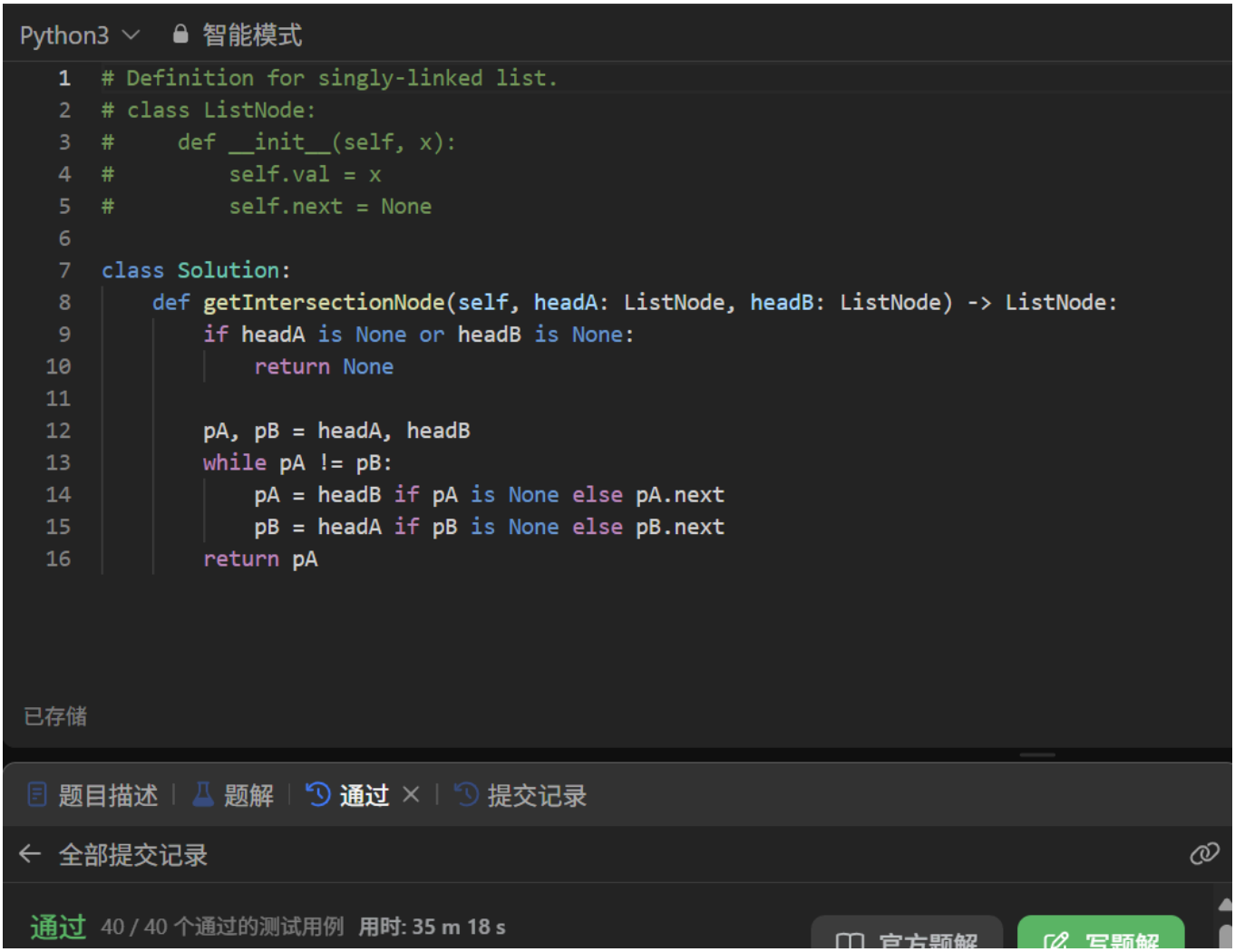
思路：

代码：

```
class Solution:

    def getIntersectionNode(self, headA: ListNode, headB: ListNode) ->
ListNode:

        if headA is None or headB is None:

            return None

        pA, pB = headA, headB

        while pA != pB:
```

```
            pA = headB if pA is None else pA.next

            pB = headA if pB is None else pB.next

        return pA
```

代码运行截图（至少包含有"Accepted"）



用时：12min左右

E206.反转链表

three pinters, recursion, https://leetcode.cn/problems/reverse-linked-list/

思路：

新建一个空链表，遍历原链表，将元素插在新链表的最左侧。

代码

```python
class Solution:

    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:

        pre = None

        cur = head

        while cur:

            nxt = cur.next

            cur.next = pre

            pre = cur

            cur = nxt

        return pre
```

（至少包含有"Accepted"）

用时10min左右

T02488: A Knight's Journey

backtracking, http://cs101.openjudge.cn/practice/02488/

思路：

代码

```
#
```

（至少包含有"Accepted"）

2. 学习总结和个人收获

字典setdefault方法非常有用，特别是在value需要是列表的时候，default=[ ]并直接跟随append很高效。

链表有一种秩序美感。另做了一道中等链表+递归：

2. 两数相加

已解答

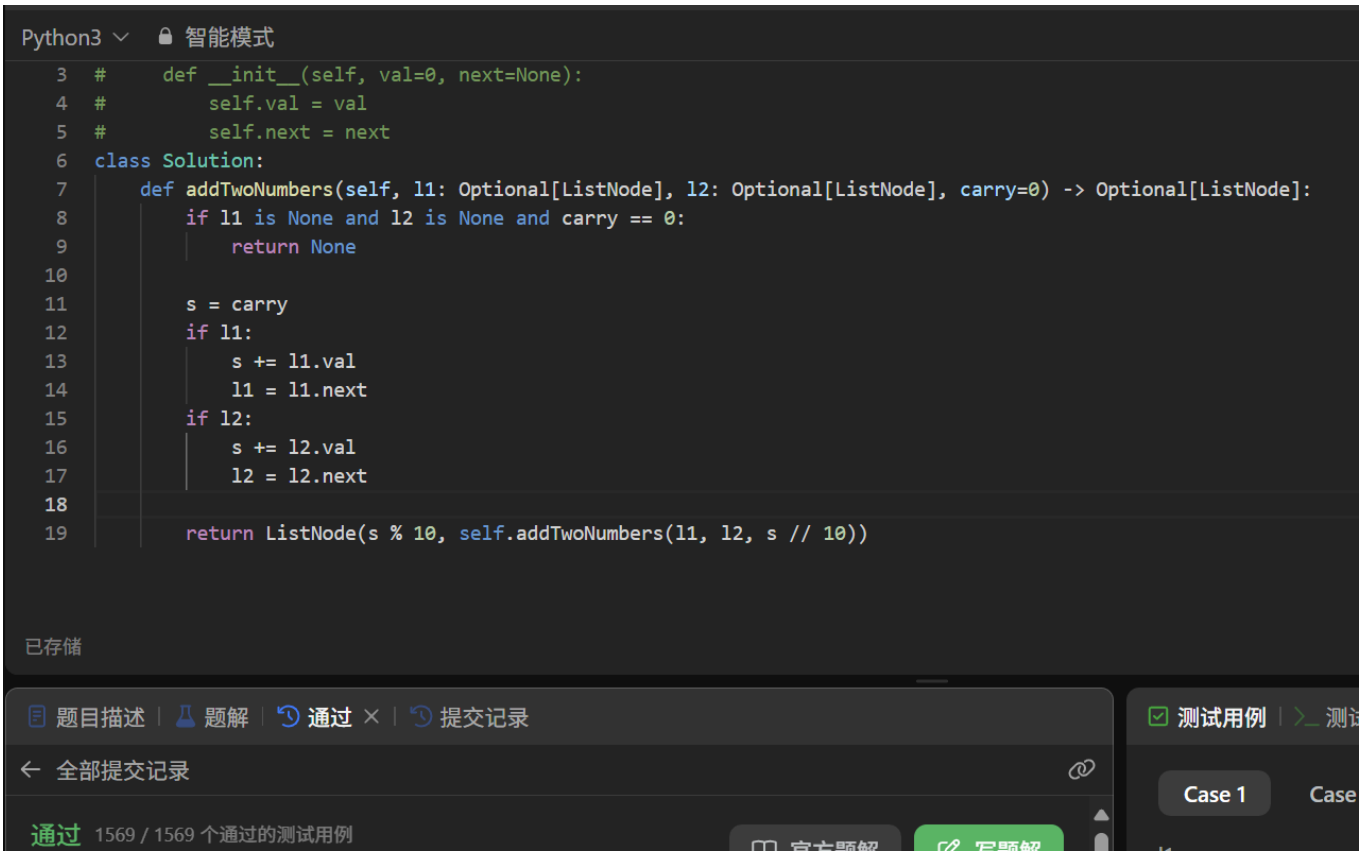
中等 | 相关标签 | 相关企业 | A文

给你两个 **非空** 的链表，表示两个非负的整数。它们每位数字都是按照 **逆序** 的方式存储的，并且每个节点只能存储 **一位** 数字。

请你将两个数相加，并以相同形式返回一个表示和的链表。

你可以假设除了数字 0 之外，这两个数都不会以 0 开头。

```
class Solution:

    def addTwoNumbers(self, l1: Optional[ListNode], l2: Optional[ListNode], carry=0) -> Optional[ListNode]:

        if l1 is None and l2 is None and carry == 0:

            return None



        s = carry

        if l1:

            s += l1.val

            l1 = l1.next

        if l2:

            s += l2.val

            l2 = l2.next



        return ListNode(s % 10, self.addTwoNumbers(l1, l2, s // 10))
```


Python3
智能模式
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def addTwoNumbers(self, l1: Optional[ListNode], l2: Optional[ListNode], carry=0) -> Optional[ListNode]:
if l1 is None and l2 is None and carry == 0:
return None
s = carry
if l1:
s += l1.val
l1 = l1.next
if l2:
s += l2.val
l2 = l2.next
return ListNode(s % 10, self.addTwoNumbers(l1, l2, s // 10))
已存储
题目描述
题解
通过
提交记录
全部提交记录
通过 1569 / 1569 个通过的测试用例
测试用例
Case 1
Case