

# Assignment #4: Matrix, 链表 & Backtracking

Updated 2226 GMT+8 Sep 29, 2025

2025 fall, Complied by 杨浩、化院

## 1. 1. 题目

### 1.1 E18161: 矩阵运算

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

请使用 @ 矩阵相乘运算符。

用时：15min

思路：

定义矩阵加法和乘法，使用@进行矩阵乘法

代码：

```

1  class Matrix():
2      def __init__(self,matrix):
3          self.matrix=matrix
4          self.rows=len(matrix)
5          self.cols=len(matrix[0])
6
7      def __add__(self,another):
8          if self.rows==another.rows and
self.cols==another.cols:
9              res=[[self.matrix[j][i]+another.matrix[j][i] for
i in range(self.cols)] for j in range(self.rows)]
10             return Matrix(res)
11         else:
12             raise ValueError
13
14     def __matmul__(self,another):
15         if self.cols==another.rows:
16             res=[[0]*another.cols for _ in range(self.rows)]
17             for i in range(another.cols):
18                 for j in range(self.rows):
19                     for k in range(self.cols):
20                         res[j][i] +=self.matrix[j]
21                         [k]*another.matrix[k][i]
22             return Matrix(res)
23         else:
24             raise ValueError
25     def __str__(self):

```

```

26         return '\n'.join(" ".join(map(str,i)) for i in
27             self.matrix)
28
29     def readin():
30         rows,cols=map(int,input().split())
31         matrix=[]
32         for i in range(rows):
33             matrix.append(list(map(int,input().split())))
34         return Matrix(matrix)
35     if __name__=='__main__':
36         a=readin()
37         b=readin()
38         c=readin()
39         try:
40             ans=a@b+c
41             print(ans)
42         except ValueError:
43             print('Error!')

```

Fence 1

代码运行截图 (至少包含有"Accepted")

#50254700提交状态

状态: Accepted 源代码	基本信息 #: 50254700 题目: E18161 提交人: Y0101 内存: 4440kB 时间: 93ms 语言: Python3 提交时间: 2025-10-08 10:21:43
---------------------	---

```

class Matrix():
    def __init__(self,matrix):
        self.matrix=matrix
        self.rows=len(matrix)
        self.cols=len(matrix[0])

    def __add__(self,another):
        if self.rows==another.rows and self.cols==another.cols:
            res=[[self.matrix[j][i]+another.matrix[j][i] for i in range(self.cols)] for j in range(self.rows)]
            return Matrix(res)
        else:
            raise ValueError

    def __matmul__(self,another):
        if self.cols==another.rows:
            res=[[0]*another.cols for _ in range(self.rows)]
            for i in range(another.cols):
                for j in range(self.rows):
                    for k in range(self.cols):
                        res[j][i] +=self.matrix[j][k]*another.matrix[k][i]
            return Matrix(res)
        else:
            raise ValueError

    def __str__(self):
        return '\n'.join(" ".join(map(str,i)) for i in self.matrix)

```

Figure 1

## 1.2 E19942: 二维矩阵上的卷积运算

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

用时: 10min

思路: 略

代码：

```

1 def jisuan(x,y,p,q):
2     for i in range(p):
3         for j in range(q):
4             res[x][y] += juanji[i][j]*matrix[i+x][j+y]
5 m,n,p,q=map(int,input().split())
6 matrix=[]
7 for i in range(m):
8     matrix.append(list(map(int,input().split())))
9 juanji=[]
10 for i in range(p):
11     juanji.append(list(map(int,input().split())))
12
13 res=[[0]*(n+1-q) for _ in range(m+1-p)]
14 for x in range(m+1-p):
15     for y in range(n+1-q):
16         jisuan(x,y,p,q)
17
18 print('\n'.join(' '.join(map(str,i)) for i in res))

```

Fence 2

代码运行截图 (至少包含有"Accepted")

#50254933提交状态

查看 提交 统计 提问

状态: Accepted

源代码	基本信息
<pre>#10: 24 def jisuan(x,y,p,q):     for i in range(p):         for j in range(q):             res[x][y] += juanji[i][j]*matrix[i+x][j+y] m,n,p,q=map(int,input().split()) matrix=[[]]</pre>	#: 50254933 题目: E19942 提交人: YO101 内存: 3652kB 时间: 23ms 语言: Python3 提交时间: 2025-10-08 10:33:55

Figure 2

## 1.3 M06640: 倒排索引

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

用时: 10min

思路:

- 注意同一篇文章中会出现相同的单词

代码：

```

1 n=int(input())
2 dic1={}
3 dic2={}

```

```

4  for i in range(n):
5      alist=input().split()
6      for j in alist[1:]:
7          if i+1 not in dic2.setdefault(j, set()):
8              dic2[j].add(i+1)
9              dic1.setdefault(j, []).append(i+1)
10
11 m=int(input())
12 for i in range(m):
13     s=input()
14     if s in dic1:
15         print(' '.join(map(str,dic1[s])))
16     else:
17         print('NOT FOUND')

```

Fence 3

代码运行截图 (至少包含有"Accepted")

#50255190提交状态

状态: Accepted	
源代码	基本信息
<pre>#10: 35 n=int(input()) dic1={} dic2={} for i in range(n):     alist=input().split()     for j in alist[1:]:         if i+1 not in dic2.setdefault(j, set()):</pre>	#: 50255190 题目: M06640 提交人: Y0101 内存: 15728kB 时间: 104ms 语言: Python3 提交时间: 2025-10-08 10:45:02

Figure 3

## 1.4 E160.相交链表

two pointers, <https://leetcode.cn/problems/intersection-of-two-linked-lists/>

思路:

- 初见时使用哈希表，后来训练一些链表题目后使用双指针

代码:

```

1 class Solution:
2     def getIntersectionNode(self, headA: ListNode, headB: ListNode) -> Optional[ListNode]:
3         ptrA, ptrB = headA, headB
4
5         while ptrA != ptrB:
6             ptrA = ptrA.next if ptrA else headB
7             ptrB = ptrB.next if ptrB else headA
8
9         return ptrA

```

Fence 4

### 代码运行截图 (至少包含有"Accepted")

通过 40 / 40 个通过的测试用例  
AND-Y 提交于 2025.08.31 10:28

官方题解

写题解

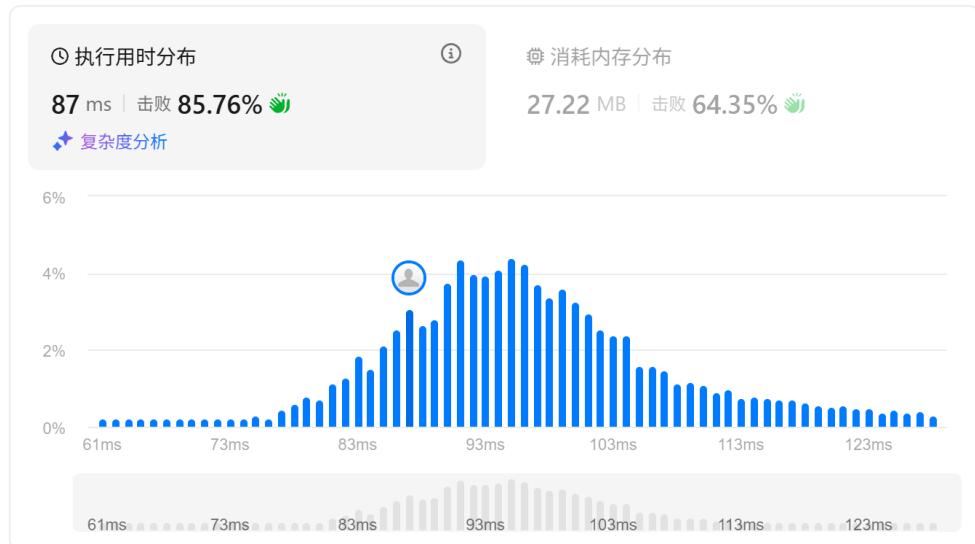


Figure 4

## 1.5 E206.反转链表

three pointers, recursion, <https://leetcode.cn/problems/reverse-linked-list/>

思路：

- 采取三指针迭代法进行处理

代码

```

1 class Solution:
2     def reverseList(self, head: Optional[ListNode]) ->
3         Optional[ListNode]:
4             if head==None or head.next==None:
5                 return head
6             node1=head
7             node2=head.next
8             head.next=None
9             node0=node1
10            while node2.next!=None:
11                node0=node1
12                node1=node2
13                node2=node1.next
14                node1.next=node0
15                if node1.next!=None:
16                    node1.next=node0
17                    node2.next=node1

```

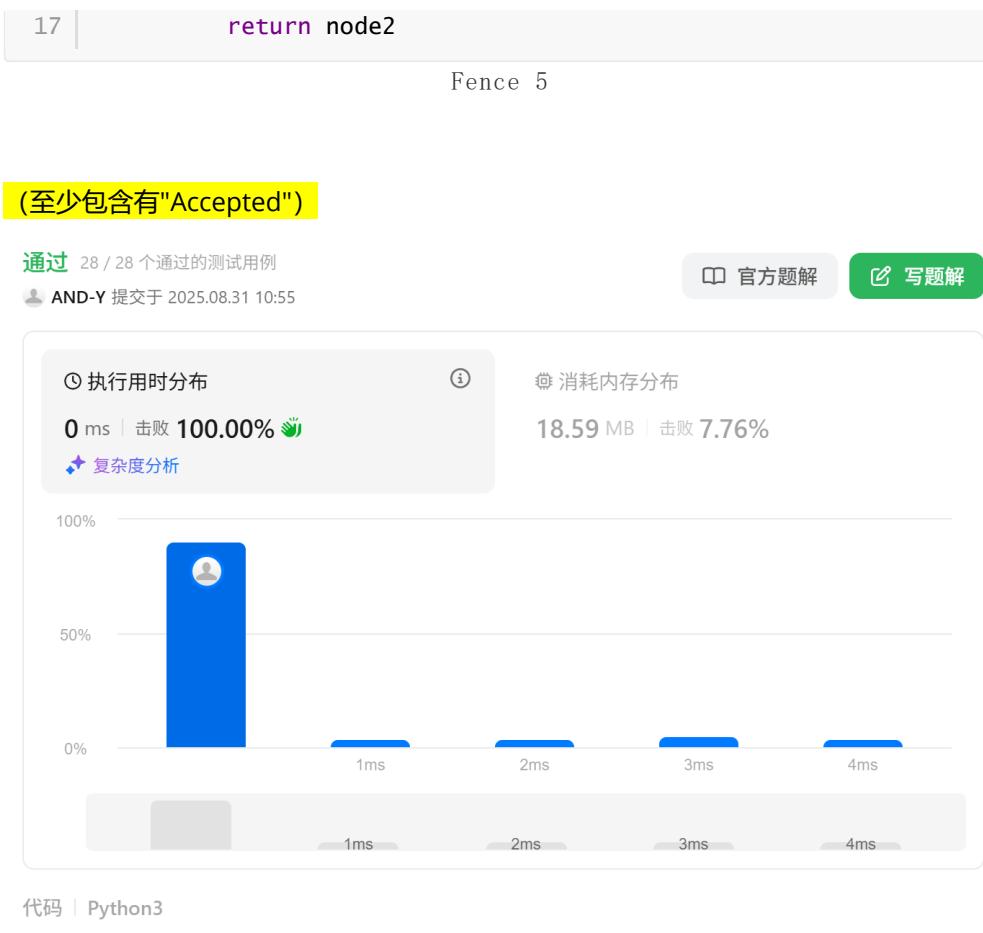


Figure 5

## 1.6 T02488: A Knight's Journey

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

思路：

- 注意字典序调整delta内部元组的顺序

代码

```

1 def dfs(hax, deep, q, p, x, y, path, ans):
2     hax[x][y]=1
3     path.append((x,y))
4     if deep==q*p-1:
5         if ans==[]:
6             ans=path[:]
7         return ans
8     for dx,dy in delta:
9         if 0<=x+dx<q and 0<=y+dy<p:
10            if hax[x+dx][y+dy]==0:
11                ans=dfs(hax, deep+1, q, p, x+dx, y+dy, path, ans)
12                if len(ans)!=0:
13                    return ans
14    hax[x][y]=0

```

```

15     path.pop()
16     return ans
17
18 n=int(input())
19 delta=[(-2,-1),(-2,1),(-1,-2),(-1,2),(1,-2),(1,2),(2,-1),
20 (2,1)]
21 for i in range(n):
22     p,q=map(int,input().split())
23     hax=[[0]*p for _ in range(q)]
24     ans=[]
25     for j in range(q):
26         for k in range(p):
27             ans=dfs(hax,0,q,p,j,k,[],ans)
28             if len(ans)!=0:
29                 break
30             if len(ans)!=0:
31                 break
32     print(f'Scenario #{i+1}:')
33     if len(ans)==0:
34         print('impossible')
35     else:
36         res=''
37         for zimu,num in ans:
38             res +=chr(zimu+65)
39             res +=str(num+1)
40         print(res)
41     if i!=n-1:
42         print()

```

Fence 6

(至少包含有"Accepted")

#49994799提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def dfs(hax,deep,q,p,x,y,path,ans):
    hax[x][y]=1
    path.append((x,y))
    if deep==q*p-1:
        if ans==[]:
            ans=path[::]
        return ans

```

## 基本信息

#: 49994799  
 题目: 02488  
 提交人: Y0101  
 内存: 4012kB  
 时间: 317ms  
 语言: Python3  
 提交时间: 2025-09-14 21:01:29

Figure 6

## 2. 2. 学习总结和个人收获

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

在LeetCode上练习递归，回溯，dfs题目，这部分的题目大部分都可以AC了，但用时普遍还较长，需要进一步训练。完成作业和每日选作时，自己写完代码AC后学习答案优化代码，同时学习一些自己不会用的函数和方法。