Assignment #4: Matrix, 链表 & Backtracking

Updated 2226 GMT+8 Sep 29, 2025

2025 fall, Complied by 胡孝齐 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： 约10min 语法题，无算法

代码：

```
row1,col1=map(int,input().split())
A=[]
for i in range(row1):
    A.append(list(map(int,input().split())))
row2,col2=map(int,input().split())
B=[]
for i in range(row2):
    B.append(list(map(int,input().split())))
row3,col3=map(int,input().split())
C=[]
for i in range(row3):
    C.append(list(map(int,input().split())))
if col1!=row2 or row1!=row3 or col2!=col3:
    print("Error!")
else:
    for i in range(row3):
```

```
        l=[]
        for j in range(col3):
            ij=C[i][j]
            for k in range(col1):
                ij+=A[i][k]*B[k][j]
            l.append(ij)
        print(*l)
```

代码运行截图 （至少包含有"Accepted"）

CS101 / 计算思维算法实践

题目 排名 状态 提问

#50218134提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
row1,col1=map(int,input().split())
A=[]
for i in range(row1):
    A.append(list(map(int,input().split())))
row2,col2=map(int,input().split())
B=[]
for i in range(row2):
    B.append(list(map(int,input().split())))
row3,col3=map(int,input().split())
C=[]
for i in range(row3):
    C.append(list(map(int,input().split())))
if col1!=row2 or row1!=row3 or col2!=col3:
    print("Error!")
else:
    for i in range(row3):
        l=[]
        for j in range(col3):
            ij=C[i][j]
            for k in range(col1):
                ij+=A[i][k]*B[k][j]
            l.append(ij)
        print(*l)
```

基本信息

#: 50218134
题目: E18161
提交人: 胡孝齐
内存: 3884kB
时间: 97ms
语言: Python3
提交时间: 2025-10-03 16:50:06

English 帮助 关于

E19942: 二维矩阵上的卷积运算

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

思路： 20min 语法题，一开始犯了两个错，一是列表符号也用的l，和索引l变量名重复了，二是l2的索引也写成了k和l导致超出范围，找了挺久的问题，最后使用debugger一下子就发现了。 之后创建变量要注意变量名重复的问题，写算法关键位置的时候要反复核对，运行时多用debugger，这样快。

代码：

```
m,n,p,q=map(int,input().strip().split())
l1=[]
for i in range(m):
    l1.append(list(map(int,input().strip().split())))

l2=[]
for i in range(p):
    l2.append(list(map(int,input().strip().split())))

for i in range(m+1-p):
    s=[]
    for j in range(n+1-q):
        aij=0
        for k in range(i,i+p):
            for l in range(j,j+q):
                aij+=l1[k][l]*l2[k-i][l-j]
```

```
            s.append(aij)
        print(*s)
```

代码运行截图 （至少包含有"Accepted"）

CS101 / 计算思维算法实践

题目 排名 状态 提问

#50218411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().strip().split())
l1=[]
for i in range(m):
    l1.append(list(map(int,input().strip().split())))

l2=[]
for i in range(p):
    l2.append(list(map(int,input().strip().split())))

for i in range(m+1-p):
    s=[]
    for j in range(n+1-q):
        aij=0
        for k in range(i,i+p):
            for l in range(j,j+q):
                aij+=l1[k][l]*l2[k-i][l-j]
        s.append(aij)
    print(*s)
```

基本信息

#: 50218411
题目: E19942
提交人: 胡孝齐
内存: 3656kB
时间: 20ms
语言: Python3
提交时间: 2025-10-03 17:22:47

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

M06640: 倒排索引

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

思路： 11min 语法题，一开始没发现索引不是最前面的数，后来一看自己测试的输出和样例不一样就改过来了

代码:

```
n=int(input().strip())
l=[]
for i in range(n):
    l.append(list(map(str,input().strip().split())))
    l[i][0]=i+1
m=int(input().strip())
for i in range(m):
    word=input().strip()
    ans=[]
    for j in range(n):
        if word in l[j]:
            ans.append(int(l[j][0]))
    if ans!=[]:
        print(*ans)
    else:
        print('NOT FOUND')
```

代码运行截图 （至少包含有"Accepted"）

CS101 / 计算思维算法实践

题目 排名 状态 提问

#50218521提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
n=int(input().strip())
l=[]
for i in range(n):
    l.append(list(map(str,input().strip().split())))
    l[i][0]=i+1
m=int(input().strip())
for i in range(m):
    word=input().strip()
    ans=[]
    for j in range(n):
        if word in l[j]:
            ans.append(int(l[j][0]))
    if ans!=[]:
        print(*ans)
    else:
        print('NOT FOUND')
```

基本信息

#: 50218521
题目: M06640
提交人: 胡孝齐
内存: 9556kB
时间: 947ms
语言: Python3
提交时间: 2025-10-03 17:38:25

English 帮助 关于

E160.相交链表

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

思路： 之前没接触过链表，看了题解，问ai了解了相关知识，扫清了疑惑。 链表类中的节点本身是地址，在比较时默认比较地址（没有类中定义相应运算符时默认比较地址），它的val属性是它的值，每一个节点指向下一个节点，构成链表，因而链表的访问只能顺序访问，但删除或添加节点是O(1)。 本题中，通过巧妙地将两个链表首尾相接的方式使得两个节点走的总路程相同，从而必然能在相交处第一次相等，当返回值为None时，无论两个链表是否一样长，都说明不相交。

代码：

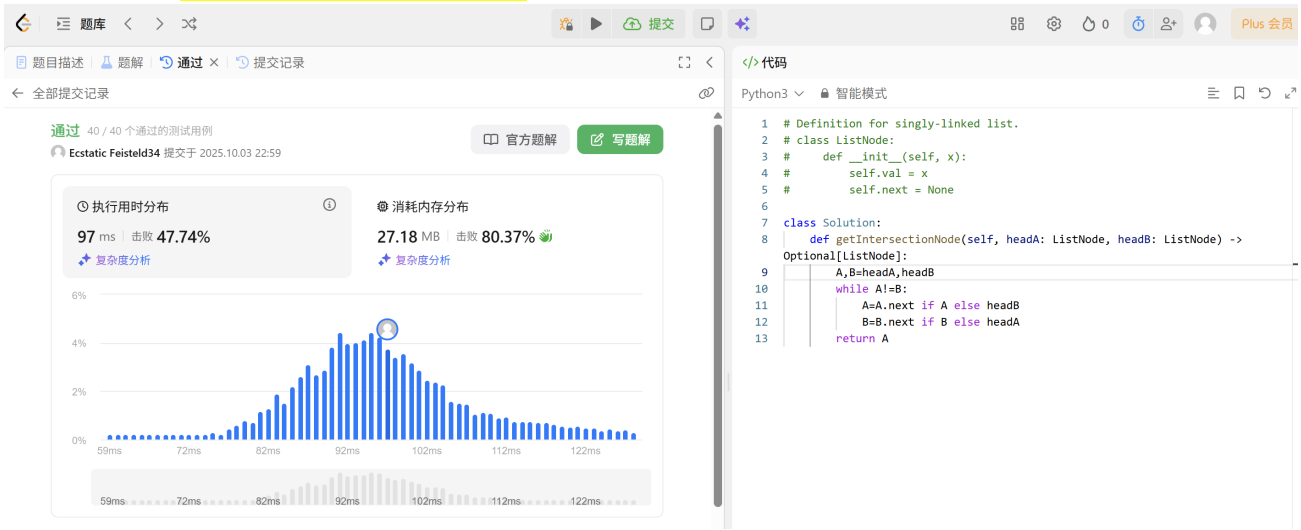
```
class Solution:
    def getIntersectionNode(self, headA: ListNode, headB: ListNode) ->
Optional[ListNode]:
        A,B=headA,headB
        while A!=B:
            A=A.next if A else headB
            B=B.next if B else headA
        return A
```

暴力解法：使用列表储存链表中的元素，暴力求解，不推荐

```
class Solution:
    def getIntersectionNode(self, headA: ListNode, headB: ListNode) ->
Optional[ListNode]:
        lA=[]
        lB=[]
        A,B=headA,headB
        while A:
            lA.append(A)
            A=A.next
        while B:
            lB.append(B)
            B=B.next
        l1=len(lA)
        l2=len(lB)
        i=0
```

```
        c=None
        while i<min(l1,l2):
            if lA[l1-i-1]==lB[l2-i-1]:
                c=lA[l1-i-1]
            i+=1
        return c
```

代码运行截图 （至少包含有"Accepted"）



E206.反转链表

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

思路：

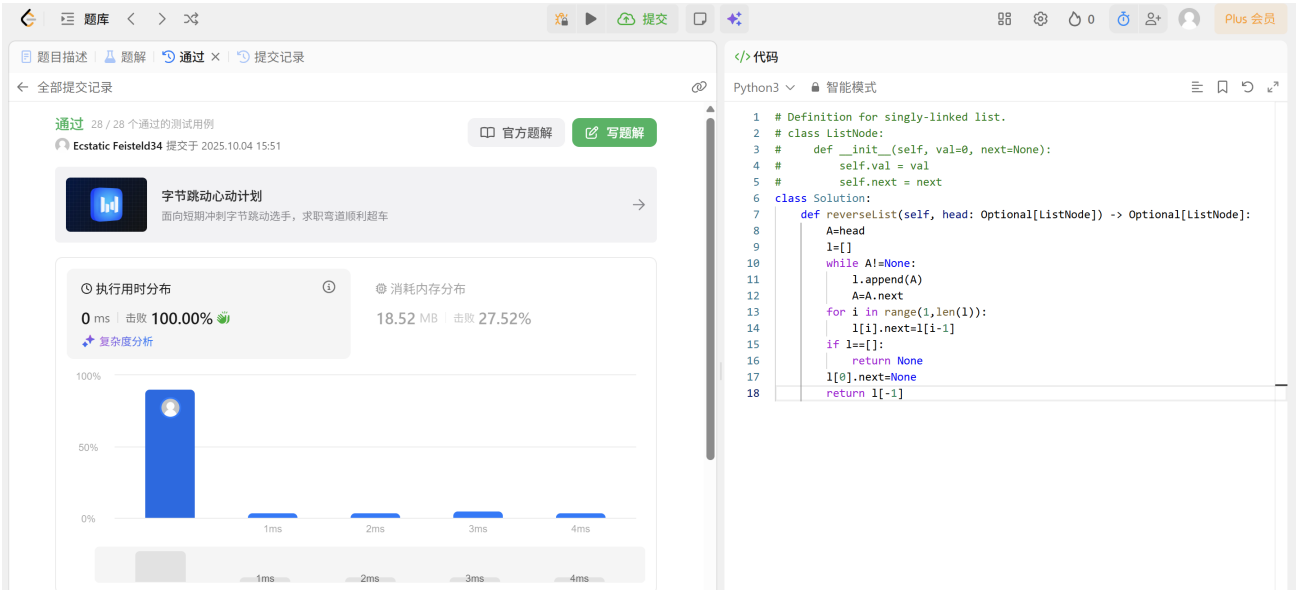
代码 暴力解法：空间复杂度较高

```
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        A=head
        l=[]
        while A!=None:
            l.append(A)
            A=A.next
        for i in range(1,len(l)):
            l[i].next=l[i-1]
        if l==[]:
            return None
        l[0].next=None
        return l[-1]
```

法二：使用变量缓存节点

```
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        b=head
        c=None
        while b!=None:
            a=b.next
            b.next=c
            c=b
            b=a
        return c
```

(至少包含有"Accepted")



T02488: A Knight's Journey

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

思路： 回溯算法，深搜，核心思想是将所有能走的下一步位置筛选出来，按顺序依次尝试（保证结果为最小字典序的路径），能够导向结束状态的返回可行，否则返回不可行， 并撤销上一步

代码

```
step=[(-2,-1),(-2,1),
      (-1,-2),(-1,2),
      (1,-2),(1,2),
      (2,-1),(2,1)]

def name(x,y):
    return chr(ord('A')+x)+str(y+1)

def judge(x,y):
    if 0<=x<q and 0<=y<p and not visited[x][y]:
        return True
    return False

def dfs(x,y):
    visited[x][y]=True
    path.append(name(x,y))
    if len(path)==p*q:
        return True
    nextstep=[]
    for dx,dy in step:
        nx=dx+x
        ny=dy+y
        if judge(nx,ny):
            nextstep.append((nx,ny))
    for nx,ny in nextstep:
        if dfs(nx,ny):
            return True
    visited[x][y]=False
    path.pop()
    return False
```

```python
n=int(input())
for i in range(n):
    p,q=map(int,input().strip().split())
    visited=[[False]*p for _ in range(q)]
    path=[]
    print(f"Scenario #{i + 1}:")
    if dfs(0,0):
        print(''.join(path))
    else:
        print('impossible')
    print('')
```

（至少包含有"Accepted"）

#50230694提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
step=[(-2,-1),(-2,1),
      (-1,-2),(-1,2),
      (1,-2),(1,2),
      (2,-1),(2,1)]

def name(x,y):
    return chr(ord('A')+x)+str(y+1)

def judge(x,y):
    if 0<=x<q and 0<=y<p and not visited[x][y]:
        return True
    return False

def dfs(x,y):
    visited[x][y]=True
    path.append(name(x,y))
    if len(path)==p*q:
        return True
    nextstep=[]
    for dx,dy in step:
        nx=dx+x
        ny=dy+y
        if judge(nx,ny):
            nextstep.append((nx,ny))
    for nx,ny in nextstep:
        if dfs(nx,ny):
            return True
    visited[x][y]=False
    path.pop()
    return False

n=int(input())
for i in range(n):
    p,q=map(int,input().strip().split())
    visited=[[False]*p for _ in range(q)]
```

基本信息

#: 50230694
题目: 02488
提交人: 胡孝齐
内存: 3700kB
时间: 241ms
语言: Python3
提交时间: 2025-10-05 12:54:21

2. 学习总结和个人收获

学习了新的数据结构链表，以及回溯算法在深搜中的使用。