

Assignment #2: 深度学习与大语言模型
Updated 2204 GMT+8 Feb 25, 2025
2025 spring, Compiled by 李振硕、信息管理系

1. 题目
18161: 矩阵运算

matrices, <http://cs101.openjudge.cn/practice/18161>

思路：

代码：

#48459717提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def cengfa(A, B):
    A_hang = len(A)
    A_lie = len(A[0])
    B_hang = len(B)
    B_lie = len(B[0])

    if A_lie != B_hang:
        return 'Error!'

    c = [[0] * B_lie for _ in range(A_hang)]

    for i in range(A_hang):
        for j in range(B_lie):
            for k in range(A_lie):
                c[i][j] += A[i][k] * B[k][j]

    return c

def jiafa(c, C):
    c_hang = len(c)
    c_lie = len(c[0])
    C_hang = len(C)
    C_lie = len(C[0])

    if c_hang != C_hang or c_lie != C_lie:
        return 'Error!'

    d = [[0] * C_lie for _ in range(c_hang)]

    for i in range(c_hang):
        for j in range(C_lie):
            d[i][j] = c[i][j] + C[i][j]

    return d

dp = []
for _ in range(3):
    row, col = map(int, input().split())
    matrix = [list(map(int, input().split())) for _ in range(row)]
    dp.append(matrix)

A, B, C = dp

def jisuanjushen(A, B, C):
    c = cengfa(A, B)
    if c == 'Error!':
        return 'Error!'

    d = jiafa(c, C)
    if d == 'Error!':
        return 'Error!'

    return d

result = jisuanjushen(A, B, C)
if result == 'Error!':
    print(result)
else:
    for row in result:
        print(' '.join(map(str, row)))
```

基本信息

#: 48459717
题目: 18161
提交人: 24n2300093007
内存: 4388kB
时间: 87ms
语言: Python3
提交时间: 2025-03-06 16:11:57

19942: 二维矩阵上的卷积运算

matrices, <http://cs101.openjudge.cn/practice/19942/>

思路：

代码：

#48462634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())
big_mt = [list(map(int, input().split())) for _ in range(m)]
small_mt = [list(map(int, input().split())) for _ in range(p)]

def cal_matrix(big_mt, small_mt):
    o_mt = [[0] * (n - q + 1) for _ in range(m - p + 1)]

    for i in range(m - p + 1):
        for j in range(n - q + 1):
            o_mt[i][j] = sum(
                big_mt[i + k1][j + k2] * small_mt[k1][k2]
                for k1 in range(p)
                for k2 in range(q)
            )

    return o_mt

matrix1 = cal_matrix(big_mt, small_mt)

print("\n".join(" ".join(map(str, row)) for row in matrix1))
```

基本信息

#: 48462634
题目: 19942
提交人: 24n2300093007
内存: 3680kB
时间: 29ms
语言: Python3
提交时间: 2025-03-06 18:30:53

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

04140: 方程求解

牛顿迭代法, <http://cs101.openjudge.cn/practice/04140/>

代码：

#48512011提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
def newton_method(f, df, x0, tol=1e-6, max_iter=100):
    x = x0
    for _ in range(max_iter):
        x_new = x - f(x) / df(x)
        if abs(x_new - x) < tol:
            return x_new
        x = x_new
    return x # 若未收敛, 则返回近似解

# 定义 f(x) 和 f'(x)
def f(x):
    return x**3 - 5*x**2 + 10*x - 80

def df(x):
    return 3*x**2 - 10*x + 10

# 选择合适的初始值
x0 = 5.0
root = newton_method(f, df, x0)
print(f"{root:.9f}")
```

基本信息

#: 48512011
题目: 04140
提交人: 24n2300093007
内存: 3528kB
时间: 33ms
语言: Python3
提交时间: 2025-03-10 16:30:12

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

06640: 倒排索引

data structures, <http://cs101.openjudge.cn/practice/06640/>

思路：

代码：

#48520263提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
# 读取文档数量
N = int(input())

# 创建倒排索引字典
inverted_index = {}

# 读取每个文档
for doc_id in range(1, N + 1):
    # 读取每个文档的单词数和单词
    data = input().split()
    ci = int(data[0]) # 文档中的单词数
    words = data[1:] # 文档中的单词列表

    # 更新倒排索引
    for word in words:
        if word not in inverted_index:
            inverted_index[word] = []
        # 如果这个单词还没有出现过该文档, 添加文档编号
        if doc_id not in inverted_index[word]:
            inverted_index[word].append(doc_id)

# 读取查询数
M = int(input())

# 对每个查询单词进行处理
for _ in range(M):
    query_word = input().strip()

    # 如果查询的单词在倒排索引中, 输出它的倒排表
    if query_word in inverted_index:
        print(" ".join(map(str, sorted(inverted_index[query_word]))))
    else:
        print("NOT FOUND")
```

基本信息

#: 48520263
题目: 06640
提交人: 24n2300093007
内存: 5140kB
时间: 86ms
语言: Python3
提交时间: 2025-03-11 15:17:02

04093: 倒排索引查询

data structures, <http://cs101.openjudge.cn/practice/04093/>

思路：

代码：

#48520594提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
def main():
    # 读取文档数量
    num_documents = int(input())

    # 存储每个文档的倒排索引
    inverted_index = {}
    for doc_id in range(1, num_documents + 1):
        parts = list(map(int, input().split()))
        inverted_index[doc_id] = set(parts[1:]) # 第一个数是文档数量, 后面

    # 读取查询数量
    num_queries = int(input())

    # 存储每个查询的条件
    queries = []
    for _ in range(num_queries):
        query_conditions = list(map(int, input().split()))
        queries.append(query_conditions)

    # 处理每个查询
    query_results = []
    for query in queries:
        result_docs = set() # 存储符合条件的文档
        exclude_docs = set() # 存储需要排除的文档
        is_first_condition = True # 标记是否是第一个必须出现的条件

        for term_id in range(num_documents):
            if query[term_id] == 1: # 必须出现的词
                if is_first_condition:
                    result_docs = inverted_index[term_id + 1].copy()
                    is_first_condition = False
                else:
                    result_docs.intersection_update(inverted_index[term_id + 1])
            elif query[term_id] == -1: # 不能出现的词
                exclude_docs.update(inverted_index[term_id + 1])

        # 排除不能出现的文档
        result_docs.difference_update(exclude_docs)
        query_results.append(result_docs)

    # 输出查询结果
    for result in query_results:
        if not result:
            print("NOT FOUND")
        else:
            print(' '.join(map(str, sorted(result))))

if __name__ == "__main__":
    main()
```

基本信息

#: 48520594

题目: 04093

提交人: 24n2300093007

内存: 9168kB

时间: 49ms

语言: Python3

提交时间: 2025-03-11 15:38:56

2. 学习总结和个人收获

<mark> “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、

这次作业的内容比较丰富，matrix、索引等等学到了很多東西，觉得我还需要多多做题。自己一个人能完成的题还是比较少。