# 项目名称：基于时变网络方法的基础设施运行效率分析

## 项目概述

本项目旨在通过对运行任务的时间、信息需求等数据进行分析，计算节点的度数、运行成本、运行效率以及网络的整体效率。项目主要包含了数据处理、矩阵运算、效率计算等多个步骤，最终输出一系列按时间排列的结果文件。

## 文件结构

### 代码文件

- `Task Driven Evaluation.py`：主程序文件，包含了从数据读取、处理到结果输出的整个流程。

### 数据文件

- `A运行任务记录.xlsx`：记录了运行任务的时间信息。

- `B运行任务的信息需求.xlsx`：记录了运行任务对应的信息需求。

- `C运行信息对应的邻接矩阵.xlsx`：包含了运行信息对应的邻接矩阵。

- `D社会节点成本记录.xlsx`：记录了社会节点的成本信息。

- `E技术节点成本记录.xlsx`：记录了技术节点的成本信息。

- `F节点运行容量赋值.xlsx`：记录了节点的运行容量信息。

- `G节点权重及属性.xlsx`：记录了节点的权重信息。

### 输出文件

- `1按时间排列的运行任务列表.csv`：第一步输出的按时间排列的运行任务列表。

- `2按时间排列的运行信息列表.csv`：第二步输出的按时间排列的运行信息列表。

- `综合邻接矩阵文件夹`：第三步输出的综合邻接矩阵文件。

- `3按时间排列的节点度数（运行信息数量）.csv`：第四步输出的按时间排列的节点度数。

- `4按时间排列的节点运行成本.csv`：第五步输出的按时间排列的节点运行成本。

- `5按时间排列的节点运行效率.csv`：第五步输出的按时间排列的节点运行效率。

- `6按时间排列的网络运行效率.csv`：第六步输出的按时间排列的网络运行效率。

## 代码流程

### 第一轮检查：前三个步骤

1. \*\*第一步：读取运行任务时间设置\*\*

- 从 `A运行任务记录.xlsx` 中读取运行任务的时间信息。

- 提取 `Code` 和 `Time` 列，处理时间范围，将每个时间点的任务存储在字典中。

- 将字典转换为 DataFrame，按时间排序并保存为 `1按时间排列的运行任务列表.csv`。

2. \*\*第二步：读取运行任务的信息需求\*\*

- 从 `B运行任务的信息需求.xlsx` 中读取运行任务的信息需求，建立映射关系。

- 读取上一步生成的 `1按时间排列的运行任务列表.csv`，获取每个时间点对应的运行信息代号。

- 输出 `2按时间排列的运行信息列表.csv`。

3. \*\*第三步：生成综合邻接矩阵\*\*

- 读取 `2按时间排列的运行信息列表.csv` 和 `C运行信息对应的邻接矩阵.xlsx`。

- 针对每个时间点的运行信息代号，提取对应的邻接矩阵并合并。

- 将合并后的矩阵保存到 `综合邻接矩阵文件夹` 中。

### 第二轮检查：四到六步

1. \*\*第四步：计算节点度数（运行信息数量）\*\*

- 读取 `综合邻接矩阵文件夹` 中的矩阵文件。

- 计算每个节点的输入输出总数，输出 `3按时间排列的节点度数（运行信息数量）.csv`。

2. \*\*第五步：计算节点运行成本和运行效率\*\*

- 读取 `3按时间排列的节点度数（运行信息数量）.csv`、`D社会节点成本记录.xlsx`、`E技术节点成本记录.xlsx` 和 `F节点运行容量赋值.xlsx`。

- 计算每个节点在每个时刻的运行成本和运行效率，分别输出 `4按时间排列的节点运行成本.csv` 和 `5按时间排列的节点运行效率.csv`。

3. \*\*第六步：计算网络整体效率\*\*

- 读取 `G节点权重及属性.xlsx` 和 `5按时间排列的节点运行效率.csv`。

- 计算每个时间点的网络整体效率，输出 `6按时间排列的网络运行效率.csv`。

### 第三轮检查：第七步

目前代码中第七步检查部分没有具体实现内容。

## 运行环境

- Python 3.x

- 依赖库：`pandas`、`networkx`、`os`

## 运行步骤

1. 确保所有数据文件（`A运行任务记录.xlsx` - `G节点权重及属性.xlsx`）都在同一目录下。

2. 运行 `Task Driven Evaluation.py` 文件。

3. 查看输出文件，了解每个步骤的结果。

## 注意事项

- 请确保数据文件的格式和内容符合代码的要求，否则可能会导致程序出错。

- 代码中部分函数（如 `calculate\_global\_efficiency`、`get\_info\_codes` 等）的具体实现被省略，需要根据实际需求进行补充。