# HW - Network Diagram

Start Assignment

- Due Dec 5 by 11:59pm
- Points 34
- Submitting a file upload
- File Types xlsm and zip

## assignment-icon.jpg HW - Network Diagram

**NOTE**: All coding problems must be done via code - no hard coding of any type is allowed for credit. Any crash or execution errors that are not gracefully handled will result in a mandatory 20% reduction. Also, all output requested should be clear with labels and easy to find, or a penalty of 10% will occur.

1. (34 pts) You are tasked with writing Python code to determine the critical path(s) for network diagrams. Network diagram configuration is given for problem 1 and problem 2 in their respective `.csv` files. The format of the CSV file is given below. The reason for the 1st row is so your program can verify the csv file is formatted for this problem hence the SER416 and the 2nd entry in that row is the version number, in this case 1.

```
1st row must be SER416, 1
2nd row is just the Column headers
1 row for each task in the network Diagram:
>   Each row then (could be in any order):  taskName, Duration, (each prede
cessor separated by commas up to the number in previous field)
```

Assumptions:

- You **CANNOT** assume that the tasks are listed in topological order.
- You **CAN** assume the network has exactly **one start** and **one terminal task**.
- You **CAN** assume a **maximum of 15 tasks** in the network.
- Task names will always be valid **strings**.

Write Python code that:

- (2 pts) Write code that prompts the user for which file to process and verifies the first line of the `.csv` file is of the format `SER416, 1`. Before continuing, The program should parse the configuration data and write it into an Excel/CSV sheet for reference.
- (10 pts) Add code that calculates and writes out the **forward and backward pass data** for each task. Specifically, compute: ( Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF) ). This

data should be written adjacent to each task's original configuration data in the output file.

- (5 pts)Add code that calculates and writes out each task's Free Float and Total Float, placed near the ES/EF/LS/LF values in the output file.
- (2 pts) Add code that calculates and writes out the **minimum total project duration** using the computed task timings and outputs it clearly in the Excel/CSV output.
- (5 pts) Add code that identifies and outputs all **tasks that are on any critical path** (i.e., tasks with zero total float). Display them clearly in the output file; there is no need to show the path structure.
- (10 pts) Write code to determine all **critical path(s)** through the network (can be more than one). Output each path sequence clearly and label it appropriately in the Excel/CSV file or terminal output.

HINT: This problem is best solved using **recursion or topological sorting** but can also be done iteratively.

**HW9-part1-prob1.csv (https://canvas.asu.edu/courses/234472/files/112833206?wrap=1) (https://canvas.asu.edu/courses/234472/files/112833206/download?download_frd=1)**

**HW9-part1-prob2.csv (https://canvas.asu.edu/courses/234472/files/112833207?wrap=1) (https://canvas.asu.edu/courses/234472/files/112833207/download?download_frd=1)**

**NOTE**: For full credit, your code must generate the correct result for the two files above and gracefully degrade if the file format header is not correct, as noted above.

# Submission:

Submit a .zip file that includes:

- network_diagram_solver.py: Your well-commented Python file.
- network_analysis_output.csv: Output of calculated task data.
- A screenshot of your output (console or CSV).

**HW Network Diagram Rubric (1)**

| Criteria | Ratings | | | | | Pts |
|---|---|---|---|---|---|---|
| File Configuration Verification<br>Code correctly reads the .csv file, verifies the first row format (SER416, 1), and writes parsed configuration to an output file. | **2 pts**<br>**Full Marks**<br>Correct file verified and configuration data saved. | | **1 pts**<br>**Partial Marks**<br>File verification partial or read error handled poorly. | | **0 pts**<br>**No Marks**<br>File not verified or failed to parse. | 2 pts |
| Forward and Backward Pass Calculation<br>Code computes Early Start, Early Finish, Late Start, and Late Finish correctly for all tasks. | **10 pts**<br>**Full Marks**<br>all data correct | **8 pts**<br>**Forward and Backward Pass**<br>Both work but some data is incorrect. | **5 pts**<br>**Forward only**<br>Forward path works with correct data and no backward path attempt | **2 pts**<br>**Partial Marks**<br>Forward path attempted by data is not all correct. | **0 pts**<br>**No Marks**<br>Not attempted or has none of the forward pass or backward pass data is correct. | 10 pts |
| Float data<br>Computes Free Float and Total Float values and includes them in the output. | **5 pts**<br>**Full Marks**<br>all values correct. | | **2.5 pts**<br>**Partial Marks**<br>Attempted but inaccurate/partial results | | **0 pts**<br>**No Marks**<br>No float values calculated or incorrect logic | 5 pts |
| Project Duration<br>Calculates total minimum project duration using latest finish among terminal tasks. | **2 pts**<br>**Full Marks**<br>Correctly computed and reported | | **1.2 pts**<br>**Partial Marks**<br>Attempted with wrong logic or result | | **0 pts**<br>**No Marks**<br>Not attempted | 2 pts |
| Identify Tasks on Critical Paths<br>Tasks with Total Float = 0 are clearly listed | **5 pts**<br>**Full Marks**<br>All correct tasks identified | | **3 pts**<br>**Partial Marks**<br>Partial list, missed or wrongly added tasks | | **0 pts**<br>**No Marks**<br>Not attempted or wrong logic | 5 pts |

| Criteria | Ratings | | | Pts |
|---|---|---|---|---|
| as being on critical path(s). | | | | |
| Critical path Identification<br>Determines all critical path sequences and reports them clearly. | **10 pts**<br>**Full Marks**<br>All paths correctly computed | **5 pts**<br>**Partial Marks**<br>Partially correct or only one critical path found | **0 pts**<br>**No Marks**<br>Incorrect logic or not attempted | 10 pts |
| | | | | Total Points: 34 |