Consider the following workflow logs:

Log 1:
- case 1 : task A
- case 2 : task A
- case 3 : task A
- case 1 : task B
- case 3 : task E
- case 1 : task C
- case 3 : task F
- case 4 : task A
- case 4 : task E
- case 3 : task H
- case 2 : task D
- case 4 : task G
- case 4 : task H
- case 2 : task C

Log 2:
- case 5 : task A
- case 6 : task A
- case 5 : task C
- case 5 : task B
- case 6 : task C
- case 6 : task D

a) Specify all the log-based ordering relations (direct successorship, causality, concurrency, exclusiveness) for Log 1.

b) Execute the $\alpha$-algorithm for Log 1 (step-by-step) and present the formal as well as the graphical representation of the resulting WF-net.

c) Assume that a second set of recorded execution sequences, given in Log 2, becomes available for the investigated process. Consider the unified log comprising all events from Log 1 and Log 2.

I. Discuss based on the ordering relations, how the result of the $\alpha$-algorithm would look like for the unified log. Show the results obtained for the unified log.

II. Elaborate on mitigation strategies for the issues observed when conducting discovery based on the unified log and apply these strategies to the given example.
**TASK 3.2**

Given are the following WF-system and workflow log.

![Workflow diagram]

Decide whether the given log is complete for the above net system. If not, explain why the log is incomplete.

**Log:**
- case 1: task A
- case 1: task C
- case 2: task A
- case 2: task D
- case 1: task E
- case 2: task B
- case 2: task E
- case 1: task B
- case 2: task F
- case 1: task F

**TASK 3.3**

Elaborate on the rediscoverability (using the \( \alpha \)-algorithms) of the following net systems. Argue by means of the ordering relations.