**Graph algorithms - directed acyclic graphs (DAGs)**

DAG example

A diagram of a network

Description automatically generated

Cycle-containing graph

A diagram of a network

Description automatically generated

Second DAG example

A diagram of a network

Description automatically generated

Issues:

1. Execute topological sorting by predecessor counting;
2. Execute topological sorting by DFS;
3. Execute both topological sorting algorithms on a graph containing a cycle.

Predecessor count sorting

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **q** | **cnt** | | | | | | |
|  |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|  | 0,5 | **0** | 1 | 2 | 2 | 1 | **0** | 2 |
| 0 | 5 | 0 | 1 | 2 | 2 | 1 | 0 | **1** |
| 5 | 4,6 | 0 | 1 | 2 | 2 | **0** | 0 | **0** |
| 4 | 6 | 0 | 1 | 2 | **1** | 0 | 0 | 0 |
| 6 | 1,3 | 0 | **0** | **1** | **0** | 0 | 0 | 0 |
| 1 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 2 | 0 | 0 | **0** | 0 | 0 | 0 | 0 |
| 2 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Scheduling problem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Act.** | **Prerequisites** | **Duration** | **Earliest** | **Latest** |
| A | E, F | 1 |  |  |
| B | E | 2 |  |  |
| C | F | 3 |  |  |
| D | A, B, C | 5 |  |  |
| E | - | 4 |  |  |
| F | - | 3 |  |  |