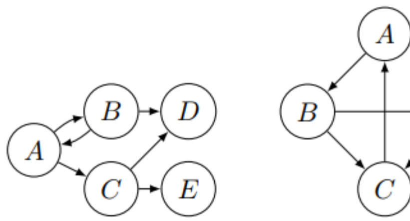
GT Lab Practical 04 - Transitive Closure in Graph

Lecturer: Pheak NEANG, Vinha CHHAENG

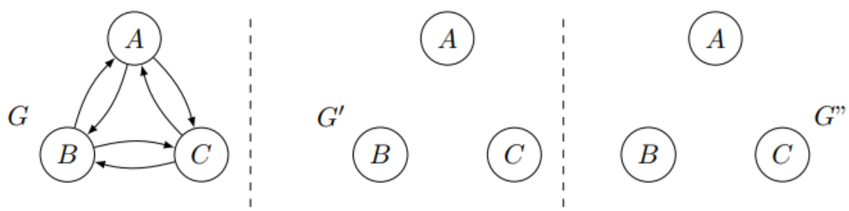
November 10, 2025

1. Definition and properties of Transitive Closure

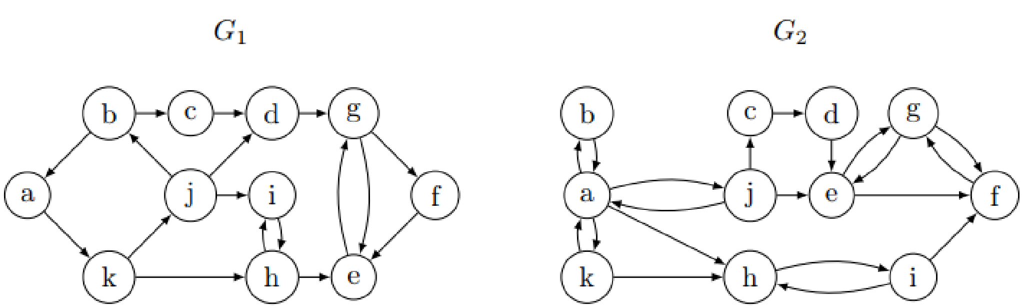
Question 1. Draw the transitive closure of the following graphs.



Question 2. Add some arcs to G' and G" such that G' and G" are τ-minimal τ-equivalent to G. Note that G" contains strictly less arcs than G'.



Question 3. Let G_1 and G_2 be the following two graphs :



(3.1). Compute the transitive closures of G_1 and G_2.

(3.2). Compute the reduced graphs G_{1R} and G_{2R} of G_1 and G_2.

(3.3). Compute the transitive closures of G_{1R} and G_{2R}.

(3.4). Remarkable for both graphs.

2. Roy Wharshall Algorithm and Matrix

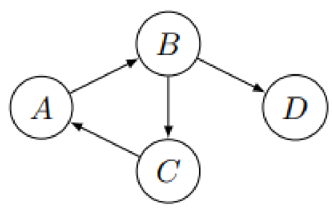
Recall that the Roy Wharshall Algorithm

Algorithm 1 Roy Wharshall Algorithm

Require: A directed graph $G = (U, A)$
Ensure: The transitive closure of G

for $w \in U$ **do**
    for $u \in U$ **do**
        for $v \in U$ **do**
            if $(u, w) \in A$ **and** $(w, v) \in A$ **then**
                Add (u, v) to A
            end if
        end for
    end for
end for

Question 1. Run the algorithm on the following graph **G** :



(1.1). Build graph using any packages in `NetworkX`.

(1.2). Check function in `NetworkX` in order to transform the graph **G** to matrix called **P**

(1.3). Build function called `Roy_Wharshall_Algorithm()` according to the pseudo-code, then apply on graph **G** is given.

(1.4). Visualization the graph by design a black color on original and blue color for new edges that obtain from transitive closure.

Question 2. Relation of Matrix and Graph

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

(2.1). Describe the algorithm using only the adjacency matrix of general directed graph.

(2.2). Build function called `Roy_Wharshall_Algorithm_Mat()` base on question **(2.1)**, then apply on matrix **M** is given by define a new matrix called **N**.

(2.3). Check function in `NetworkX` in order to transform the matrix **N** to graph called **T**

(2.4). Visualization the graph by design a black color on original and blue color for new edges that obtain from transitive closure.

3. Problem for equipment of a workshop

Nine machines, $a, b..., h$ and i, are installed in a workshop. Mechanical parts are produced by successive machines (drills, polishers, welds, . . .). We want to build conveyor belts at minimum cost in order to move the pieces from a machine to another. The following array gives, for each machine M, which machines may be used in a next step to build a product. We must link M to each of those machines by a succession of belts. Model this problem with a graph problem and solve it.

Machine	**Next Machines**
a	b, c, d, e, f, g, h, i
b	a, c, d, e, f, g, h, i
c	d, e
d	e
e	d
f	d, e, g, h, i
g	d, e, f, h, i
h	d, e, f, g, i
i	d, e, f, g, h

Table 1: Machines and their possible next-step connections

(Q1). Model the given workshop setup as a **directed graph**, where each node represents a machine and each directed edge represents a possible conveyor connection from one machine to its next-step machines. Clearly define the adjacency list or adjacency matrix corresponding to this graph.

(Q2). Determine which **graph algorithm** should be applied to minimize the total number of conveyor belts (or the total connection cost) required to connect all machines. Justify your choice of algorithm.

(Q3). Implement in file named `machine.py`, ensuring that it computes the optimal configuration (minimum-cost connection) for the given network of machines.

Note

- Summit only file `machine.py`
- For both files `machine.py` you need created by yourself.