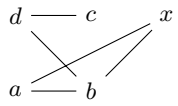
Graphs. Connectivity. Flow networks

Graph connectivity. Transitive closures

1. Consider the undirected graph G with the graphical representation



What are the representations of this graph with

(a) List of edges and list of nodes.

(b) Lists of adjacencies.

(c) Adjacency matrix A_G.

Then

(d) Compute A_G^3 and decide which nodes can be reached from c with a path of length 3.

Answer

(a) $E = [\{a, b\}, \{a, x\}, \{b, d\}, \{b, x\}, \{c, d\}]$
$V = [a, b, c, d, x]$

(b) $a \mapsto [b, x]$ $\quad d \mapsto [d, x]$
$b \mapsto [a, d, x]$ $\quad x \mapsto [a, b]$
$c \mapsto [d]$

(c) We assume that the nodes are enumerated in the order $[a, b, c, d, x, y]$. In this case, the adjacency matrix A_G looks as follows:

	a	b	c	d	x	y
a	0	1	0	0	1	0
b	1	0	0	1	1	0
c	0	0	0	1	0	0
d	0	1	1	0	0	0
x	1	1	0	0	0	0
y	0	0	0	0	0	0

thus $A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

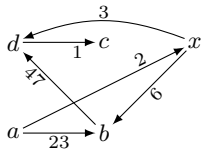
(d) $A_G^2 = \begin{pmatrix} 0&1&0&0&1&0\\ 1&0&0&1&1&0\\ 0&0&0&1&0&0\\ 0&1&1&0&0&0\\ 1&1&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix} \odot \begin{pmatrix} 0&1&0&0&1&0\\ 1&0&0&1&1&0\\ 0&0&0&1&0&0\\ 0&1&1&0&0&0\\ 1&1&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix} = \begin{pmatrix} 1&1&0&1&1&0\\ 1&1&1&0&1&0\\ 0&1&1&0&0&0\\ 1&0&0&1&1&0\\ 1&1&0&1&1&0\\ 0&0&0&0&0&0 \end{pmatrix}$

$A_G^3 = A_G^2 \odot A_G = \begin{pmatrix} 1&1&0&1&1&0\\ 1&1&1&0&1&0\\ 0&1&1&0&0&0\\ 1&0&0&1&1&0\\ 1&1&0&1&1&0\\ 0&0&0&0&0&0 \end{pmatrix} \odot \begin{pmatrix} 0&1&0&0&1&0\\ 1&0&0&1&1&0\\ 0&0&0&1&0&0\\ 0&1&1&0&0&0\\ 1&1&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix} =$

$$\begin{pmatrix} 1&1&1&1&1&0\\ 1&1&0&1&1&0\\ 1&0&0&1&1&0\\ 1&1&1&0&1&0\\ 1&1&1&1&1&0\\ 0&0&0&0&0&0 \end{pmatrix}.$$

c is the third node, and the third line of the matrix A_G^3 is $[1, 0, 0, 1, 1, 0]$, which has nonzero entries at positions 1, 4 and 5. Thus c is connected with paths of length 3 to the first, 4-th, and 5-th node, i.e., to a, d and x.

2. Consider the weighted digraph G with the graphical rpresentation



Indicate the representations of this graph with

(a) List of edges and list of nodes.

(b) Lists of adjacencies.

(c) Adjacency matrix A_G.

(d) Weight matrix M_G.

and then

(e) Compute A_G^4 and indicate the pairs of nodes connected by paths of length 4.

Answer:

(a) $E = [(a,b),(a,x),(b,d),(d,c),(x,b),(x,d)]$,
$V = [a,b,c,d,x]$

(b) $a \mapsto [b,x]$ $\quad d \mapsto [c]$
$b \mapsto [d]$ $\quad x \mapsto [b,d]$
$c \mapsto []$

(c) If we fix the enumeration $[a,b,c,d,x]$ for the nodes of the graph, then

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

Note that these three representations of G (with list of edges + list of nodes, with lists of adjacencies, and with adjacency matrix) are incomplete because they do not store the weights of arcs.

(d) If we fix the enumeration $[a,b,c,d,x]$ for the nodes of the graph, then

$$W_G = \begin{pmatrix} 0 & 23 & \infty & \infty & 2 \\ \infty & 0 & \infty & 47 & \infty \\ \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & 1 & 0 & \infty \\ \infty & 6 & \infty & 3 & 0 \end{pmatrix}.$$

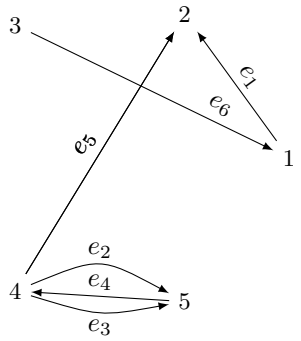
(e) $A_G^2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$

$A_G^4 = A_G^2 \odot A_G^2 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

The only nonzero element of A_G^4 is at position (1,3) $\Rightarrow$ there is only pair of nodes between whom there is a path of length 4 in G: from the 1st node (which is a) to the 3rd node (which is c).

For example, such a path is $[a,x,b,d,c]$.

3. Let G be the digraph illustrated below:



Specify the representations of G with

(a) Adjacency matrix A_G.

(b) Incidence matrix M_G.

ANSWER:

(a) For the enumeration $[1, 2, 3, 4, 5]$ of nodes, the adjacency matrix is

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

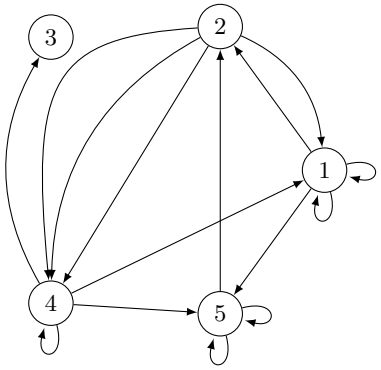
(b) For the enumerations $[1, 2, 3, 4, 5]$ of nodes, and $[e_1, e_2, e_3, e_4, e_5, e_6]$ of edges, the incidence matrix M_G of this graph is

	e_1	e_2	e_3	e_4	e_5	e_6
1	-1	0	0	0	0	-1
2	1	0	0	0	1	0
3	0	0	0	0	0	1
4	0	-1	-1	1	-1	0
5	0	1	1	-1	0	0

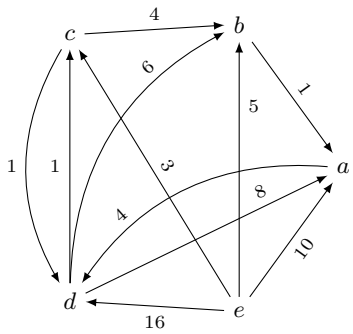
4. Draw a digraph G whose adjacency matrix is

$$A_G = \begin{pmatrix} 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 2 \end{pmatrix}$$

ANSWER: A_G has dimension 5×5, thus G has 5 nodes $\Rightarrow$ we can assume that the list of nodes is $[1, 2, 3, 4, 5]$.
We draw these nodes in the plane, and afterwards we draw arcs between nodes in accordance with the element values of the adjacency matrix A_G:



5. For the weighted digraph G depicted below



Use Warshall's algorithm to comptue the lightest paths between all pairs of nodes in this graph.

Answer: First, we fix a traversal order for the nodes of G: $[a, b, c, d, e]$. Thus a is node 1, b is 2, c is 3, d is 4, and e is 5.

To simplify the description of how the algorithm works, I will write in one matrix $WP^{[k]}$ the element values of both matrices $W^{[k]}$ and $P^{[k]}$:

- $WP^{[0]} = \begin{pmatrix} [a]_0 & \bullet_\infty & \bullet_\infty & [a,d]_4 & \bullet_\infty \\ [b,a]_1 & [b]_0 & \bullet_\infty & \bullet_\infty & \bullet_\infty \\ \bullet_\infty & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet_\infty \\ [d,a]_8 & [d,b]_6 & [d,c]_1 & [d]_0 & \bullet_\infty \\ [e,a]_{10} & [e,b]_5 & [e,c]_3 & [e,d]_{16} & [e]_0 \end{pmatrix}$

  The red marked indices represent the values of $W^{[0]}$ at he respective positions; if we elide the indices, we obtain the matrix $P^{[0]}$.
  From now on we will omit the indices of $\bullet$, because the values of $W^{[k]}$ at the positions where $\bullet$ occurs are always ∞.
- For the computation of $WP^{[1]}$ we check if we can get lighter paths if we go through node 1 (which is a):

$$WP^{[1]} = \begin{pmatrix} [a]_0 & \bullet & \bullet & [a,d]_4 & \bullet \\ [b,a]_1 & [b]_0 & \bullet & [b,a,d]_5 & \bullet \\ \bullet & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet \\ [d,a]_8 & [d,b]_6 & [d,c]_1 & [d]_0 & \bullet \\ [e,a]_{10} & [e,b]_5 & [e,c]_3 & [e,a,d]_{14} & [e]_0 \end{pmatrix}$$

The blue entries are those that got modified (they are paths that are lighter than the previous ones).

- For the computation of $WP^{[2]}$ we check if we can get lighter paths if we go through node 2 (which is b):

$$WP^{[2]} = \begin{pmatrix} [a]_0 & \bullet & \bullet & [a,d]_4 & \bullet \\ [b,a]_1 & [b]_0 & \bullet & [b,a,d]_5 & \bullet \\ [c,b,a]_5 & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet \\ [d,b,a]_7 & [d,b]_6 & [d,c]_1 & [d]_0 & \bullet \\ [e,b,a]_6 & [e,b]_5 & [e,c]_3 & [e,b,a,d]_{10} & [e]_0 \end{pmatrix}$$

- For the computation of $WP^{[3]}$ we check if we can get lighter paths if we go through node 3 (which is c):

$$WP^{[3]} = \begin{pmatrix} [a]_0 & \bullet & \bullet & [a,d]_4 & \bullet \\ [b,a]_1 & [b]_0 & \bullet & [b,a,d]_5 & \bullet \\ [c,b,a]_5 & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet \\ [d,c,b,a]_6 & [d,c,b]_5 & [d,c]_1 & [d]_0 & \bullet \\ [e,b,a]_6 & [e,b]_5 & [e,c]_3 & [e,c,d]_4 & [e]_0 \end{pmatrix}$$

- For the computation of $WP^{[4]}$ we check if we can get lighter paths if we go through node 4 (which is d):

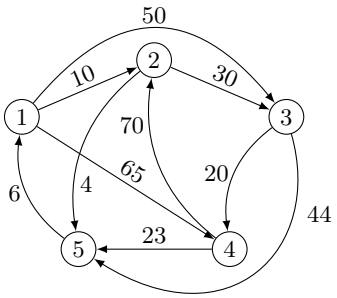
$$WP^{[4]} = \begin{pmatrix} [a]_0 & [a,d,c,b]_9 & [a,d,c]_5 & [a,d]_4 & \bullet \\ [b,a]_1 & [b]_0 & [b,a,d,c]_6 & [b,a,d]_5 & \bullet \\ [c,b,a]_5 & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet \\ [d,c,b,a]_6 & [d,c,b]_5 & [d,c]_1 & [d]_0 & \bullet \\ [e,b,a]_6 & [e,b]_5 & [e,c]_3 & [e,c,d]_4 & [e]_0 \end{pmatrix}$$

- For the computation of $WP^{[5]}$ we check if we can get lighter paths if we go through node 5 (which is e):

$$WP^{[5]} = WP^{[4]} = \begin{pmatrix} [a]_0 & [a,d,c,b]_9 & [a,d,c]_5 & [a,d]_4 & \bullet \\ [b,a]_1 & [b]_0 & [b,a,d,c]_6 & [b,a,d]_5 & \bullet \\ [c,b,a]_5 & [c,b]_4 & [c]_0 & [c,d]_1 & \bullet \\ [d,c,b,a]_6 & [d,c,b]_5 & [d,c]_1 & [d]_0 & \bullet \\ [e,b,a]_6 & [e,b]_5 & [e,c]_3 & [e,c,d]_4 & [e]_0 \end{pmatrix}$$

The elements of $WP^{[5]}$ are the lightest paths between the corresponding nodes, and their subscripts represent their corresponding weights.

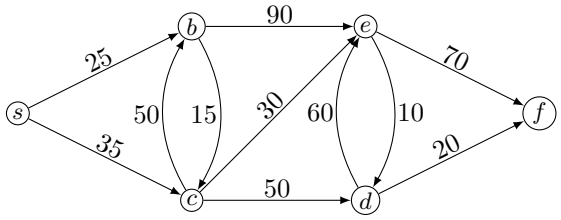
Homework 1: Use Warshall's algorithm to compute the lightest paths between any pair of nodes in the weighted directed graph depicted below:



Dijkstra's Algorithm

Allows to compute efficiently the lightest paths from a source node to all other nodes in a weighted digraph.

Homework 2: The following picture illustrates a weighted digraph with source node s, and also the values of π and d associated with the nodes of the graph by the initialisation step of Dijkstra's algorithm:



Compute the values of π and d for the nodes of the graph after every **while** loop that marks a node in Dijkstra's algorithm. Afterwards, draw the rooted tree G_π.

Ford-Fulkerson's algorithm

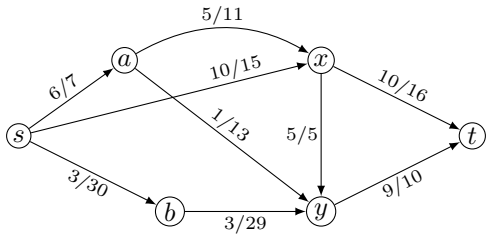
The purpose of this exercise is to illustrate how to use Ford-Fulkerson's algorithm to compute a maximum flow from a source to a destination in a flow network .

Remember that:

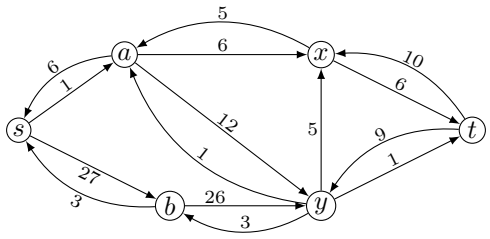
- For any representation $G + f$ of a flow network G with a ntework flow f, we construct the **residual network** G_f as follows:
  - ▶ Se înlocuieşte fiecare arc $x \xrightarrow{a/c} y$ din $G + f$ cu două arce: $x \overset{c-a}{\underset{a}{\rightleftarrows}} y$
    Remember that in $G + f$, an arc $x \xrightarrow{0/c} y$ has the simplified representation $x \xrightarrow{c} y$, and in G_f we do not draw arcs with label 0.

- At every step, Ford-Fulkerson's algorithm looks for an augmenting path in the residual network G_f, as follows:
  1. We verify if there is a path $s \rightsquigarrow t$ from source s to destination t in G_f. This test is performed as follows:
     - Perform a breadth-first traversal of G_f, starting from source s $\Rightarrow$ the rooted tree $\texttt{Bft}_{G_f}$.
       The shorthand `Bft` is derived from *breadth-first traversal*.
     - If $\texttt{Bft}_{G_f}$ **has a branch from** s **to** t, we choose the augmenting path $s \overset{f'}{\rightsquigarrow} t$ along that branch, with the flow f' whose value is equal with the minimum capacity of an arc along that path.
     - If $\texttt{Bft}_{G_f}$ **has no branch from** s **to** t, the algorithm stops and decides that $G + f$ is a flow network with maximum flow f.
  2. If the previous step found an augmenting path $s \overset{f'}{\rightsquigarrow} t$, construct $G + (f + f')$ and **goto** step 1.

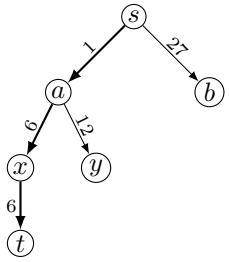
For example, if $G + f$ is



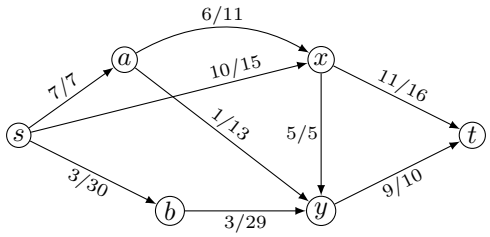
then G_f is



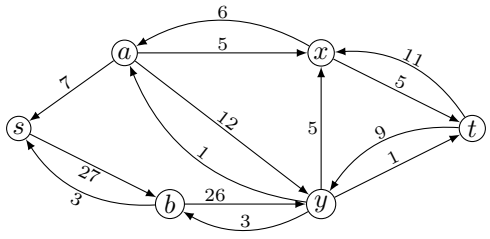
and $\texttt{Bft}_{G_f}$ can be (the exact shape of $\texttt{Bft}_{G_f}$ depends on the traversal order of the neighbours of a node):



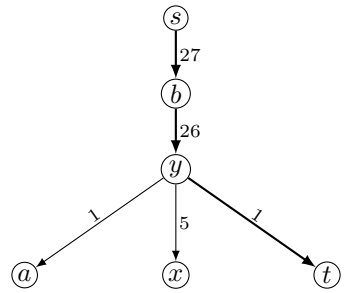
We get the augmenting path $s \xrightarrow{1} a \xrightarrow{6} b \xrightarrow{6} x \xrightarrow{6} t$ with a flow whose value is 1. After adding f' to f, we obtain $G + f_1$ where $f_1 = f + f'$, which looks as follows:



In the next round, G_{f_1} is



and $\texttt{Bft}_{G_{f_1}}$ is the rooted tree



$\Rightarrow$ the augmenting path $s \xrightarrow{27} b \xrightarrow{26} y \xrightarrow{1} t$ whose flow value is 1, a.s.o. ...

Homework 3: Use Ford-Fulkerson's algorithm to find a maximum flow in the flow network