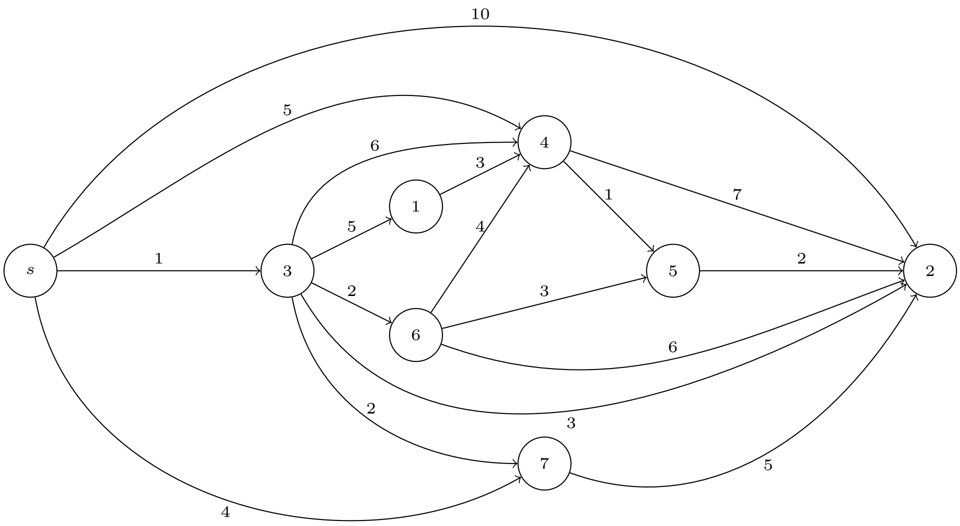
Optimization of Logistics Systems / Logistics Systems Planning I
2. Exercise

Exercise 2.1 (Topological sorting)

Given the digraph $D = (V, A)$. Sort the nodes of the digraph topologically.



Solution topological sorting

- Prerequisite: graph is acyclic
- Procedure:
  1. Set $V' = V$, List $X = (\cdot)$
  2. Stop if $V' = \emptyset$
  3. Calculate indegrees $d^-(i)\ \forall i \in V'$
  4. Select arbitrary node $v \in V'$ with $d^-(v) = 0$ (if there is no v such that $d^-(v) = 0$, the graph is cyclic)
  5. Remove v from V' (and all arcs that are incident with v), add v to the end of X, goto step 2.

V'	Indegrees in $D(V')$								X
	S	1	2	3	4	5	6	7	
$\{S, 1, 2, \dots, 7\}$	0	1	6	1	4	2	1	2	(S)
$\{1, 2, 3, 4, 5, 6, 7\}$	-	1	$\underline{5}$	$\underline{0}$	$\underline{3}$	2	1	$\underline{1}$	$(S, 3)$
$\{1, 2, 4, 5, 6, 7\}$	-	$\underline{0}$	$\underline{4}$	-	$\underline{2}$	2	$\underline{0}$	$\underline{0}$	$(S, 3, 6)$
$\{1, 2, 4, 5, 7\}$	-	0	$\underline{3}$	-	$\underline{1}$	$\underline{1}$	-	0	$(S, 3, 6, 1)$
$\{2, 4, 5, 7\}$	-	-	3	-	$\underline{0}$	1	-	0	$(S, 3, 6, 1, 4)$
$\{2, 5, 7\}$	-	-	$\underline{2}$	-	-	$\underline{0}$	-	0	$(S, 3, 6, 1, 4, 5)$
$\{2, 7\}$	-	-	1	-	-	-	-	0	$(S, 3, 6, 1, 4, 5, 7)$
$\{2\}$	-	-	$\underline{0}$	-	-	-	-	-	$(S, 3, 6, 1, 4, 5, 7, 2)$

$\Rightarrow$ a top. sorting is given by $(S, 3, 6, 1, 4, 5, 7, 2)$.

Exercise 2.2 (Pulling algorithm)

Use the pulling algorithm to determine the length of the shortest paths from node s to all other nodes in the digraph of Exercise 2.1.

Solution:

Pulling algorithm: ("From which nodes can node i be reached?")

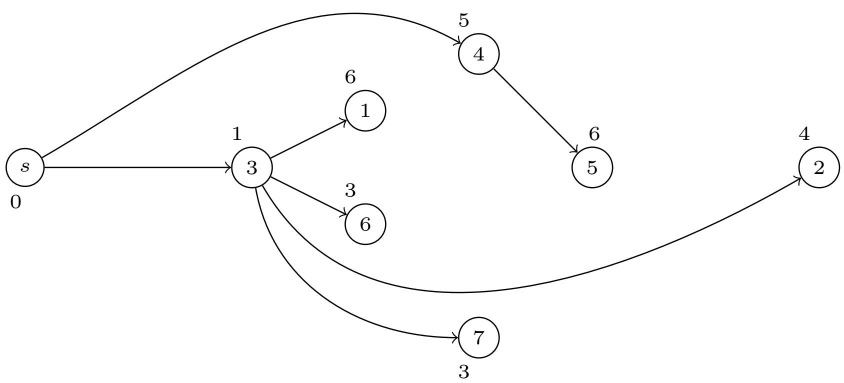
- Prerequisites: digraph is acyclic, arbitrary arc weights, nodes are given in topological sorting X
- Procedure:
  1. Select s according to sorting X; $d(s) = 0$, $pred(s) = 0$, $V' = V \backslash \{s\}$
  2. Stop if $V' = \emptyset$.
  3. Select next node $j \in V'$ according to sorting X, $V' = V' \backslash \{j\}$
  4. Calculate $d(i) + c_{ij}$ for all predecessors of j (arcs in backward star $\delta^-(j)$) and select the i with smallest $d(i) + c_{ij}$
  5. Set $d(j) = d(i) + c_{ij}$, $pred(j) = i$, goto Step 2.

  ($\rightarrow$ Label of a node is set when the labels of all potential predecessors have been set.)

Topological sorting $X = (S, 3, 6, 1, 4, 5, 7, 2)$

Node j	Backward star $\delta^-(j)$	$d(i) + c_{ij}$	min	Labels	
Initialization				$d(s) = 0$	$pred(s) = 0$
3	$(s, 3)$	1	1	$d(3) = 1$	$pred(3) = s$
6	$(3, 6)$	3	3	$d(6) = 3$	$pred(6) = 3$
1	$(3, 1)$	6	6	$d(1) = 6$	$pred(1) = 3$
4	$(s, 4), (1, 4), (3, 4), (6, 4)$	$5, 9, 7, 7$	5	$d(4) = 5$	$pred(4) = s$
5	$(4, 5), (6, 5)$	$6, 6$	6	$d(5) = 6$	$pred(5) = 4$
7	$(s, 7), (3, 7)$	$4, 3$	3	$d(7) = 3$	$pred(7) = 3$
2	$(s, 2), (3, 2), (4, 2), (5, 2), (6, 2), (7, 2)$	$10, 4, 12, 8, 9, 8$	4	$d(2) = 4$	$pred(2) = 3$

$\Rightarrow$ Shortest-path-tree:



Exercise 2.3 (Reaching algorithm)

Use the reaching algorithm to determine the length of the shortest paths from node s to all other nodes in the digraph of Exercise 2.1.

Solution:

Reaching algorithm: ("Which nodes can be reached from node i?")

- Prerequisites: digraph is acyclic, arbitrary arc weights, nodes are given in topological sorting X
- Solution method:
  1. Select s according to sorting X; $d(s) = 0$, $pred(s) = 0$, $d(i) = \infty \ \forall i \in V \setminus \{s\}$, $V' = V \setminus \{s\}$
  2. Stop if $V' = \emptyset$.
  3. Select next node $i \in V'$ according to sorting X, $V' = V' \setminus \{i\}$
  4. Check for all successors j from i (arcs in forward star $\delta^+(i)$) whether $d(j) > d(i) + c_{ij}$, if yes, set $d(j) = d(i) + c_{ij}$ and $pred(j) = i$, goto Step 2.

  (Note: New paths are investigated from ultimately marked nodes.)

Topological sorting $X = (S, 3, 6, 1, 4, 5, 7, 2)$

Node i	Forward star $\delta^+(i)$	$d(i) + c_{ij}$	Labels of improved nodes
Initialization			$d(s) = 0$ $pred(s) = 0$
s	$(s, 2), (s, 3), (s, 4), (s, 7)$	$10, 1, 5, 4$	$d(2) = 10$ $pred(2) = s$, $d(3) = 1$ $pred(3) = s$, $d(4) = 5$ $pred(4) = s$, $d(7) = 4$ $pred(7) = s$
3	$(3, 1)(3, 2)(3, 4)(3, 6)(3, 7)$	$6, 4, 7, 3, 3$	$d(1) = 6$ $pred(1) = 3$, $d(2) = 4$ $pred(2) = 3$, $d(6) = 3$ $pred(6) = 3$, $d(7) = 3$ $pred(7) = 3$
6	$(6, 2), (6, 4), (6, 5)$	$9, 7, 6$	$d(5) = 6$ $pred(5) = 6$
1	$(1, 4)$	9	-
4	$(4, 2), (4, 5)$	$12, 6$	-
5	$(5, 2)$	8	-
7	$(7, 2)$	8	-
2	-	-	-

$\Rightarrow$ Shortest path tree: (Note: Different optimal solution, see node 5!)