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

Exercise 3.1 (Longest path)

A *longest path* from s to t is a path from s to t for which the sum of the arc weights is maximum.

1. How can you transform the problem of the longest paths in a given acyclic weighted digraph $D = (V, A, d_{ij})$ into a corresponding shortest path problem?
2. Solve the longest path problem for the graph from Exercise 2.1.
3. Does your approach work for any digraph?

Solution:

1. Use property: $\min(f(x)) = -\max(-f(x))$
   - For the shortest path problem: $\min \sum_{(i,j)\in A} c_{ij}x_{ij} = -\max \sum_{(i,j)\in A} (-c_{ij})x_{ij}$ (or for the longest path problem: $\max \sum_{(i,j)\in A} c_{ij}x_{ij} = -\min \sum_{(i,j)\in A} (-c_{ij})x_{ij}$)
   - Transform weighted digraph D from Exercise 2.1 into weighted digraph $D' = (V, A, c_{ij})$ with $c_{ij} := -d_{ij},\ \forall (i,j) \in A$
   - Then use pulling or reaching algorithm
2. Pulling algorithm:
   - Topological sorting: $(s, 3, 6, 1, 4, 5, 7, 2 = t)$
   - Solution steps:

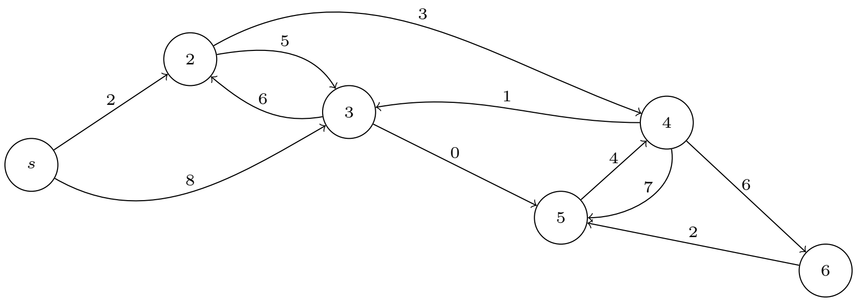
Node j	Backward star $\delta^-(j)$	$d(i) + c_{ij}$	min	Labels	
Initialization				$d(s) = 0$	$pred(s) = undef$
3	s	-1	-1	$d(3) = -1$	$pred(3) = s$
6	3	-3	-3	$d(6) = -3$	$pred(6) = 3$
1	3	-6	-6	$d(1) = -6$	$pred(1) = 3$
4	$s, 1, 3, 6$	$-5, -9, -7, -7$	-9	$d(4) = -9$	$pred(4) = 1$
5	$4, 6$	$-10, -6$	-10	$d(5) = -10$	$pred(5) = 4$
7	$s, 3$	$-4, -3$	-4	$d(7) = -4$	$pred(7) = s$
2	$s, 3, 4, 5, 6, 7$	$-10, -4, -16, -12, -9, -9$	-16	$d(2) = -16$	$pred(2) = 4$

   $\Rightarrow$ longest path from s to 2: $(s, 3, 1, 4, 2)$ with cost 16
3. Approach is only possible on graphs which do not contain cycles with positive weights.

Exercise 3.2 (Shortest path problem with positive arc weights)

Given the following weighted digraph $D = (V, A, d_{ij})$:

1. Solve the shortest path problem starting from node s with the help of the Dijkstra algorithm.
2. What does the shortest-path-tree associated with your solution look like?



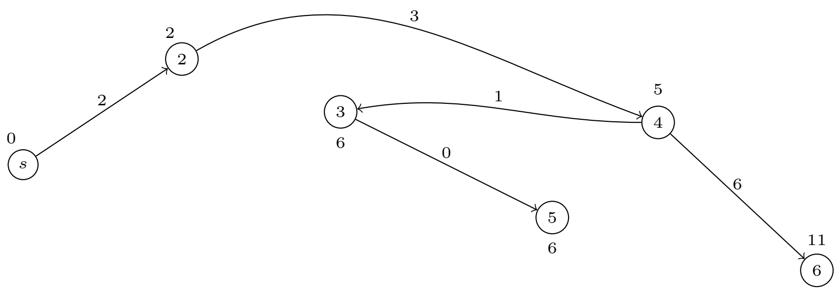
Solution:

Dijkstra algorithm:

- Prerequisite: $d_{ij} \geq 0, \ \forall (i,j) \in A$
- Procedure:
  1. $d(s) = 0$, $pred(s) = undef$, ls temporarily marked, $d(j) = \infty$ for all other nodes
  2. If no node is temporarily labeled: stop.
  3. Select the temporarily marked node i with *smallest* $d(i)$; i becomes ultimately marked
  4. Check for all successors j of i (that are not ultimately marked) whether: $d(j) > d(i) + c_{ij}$. If yes, set $d(j) = d(i) + c_{ij}$, $pred(j) = i$ and set j as temporarily marked, goto step 2.

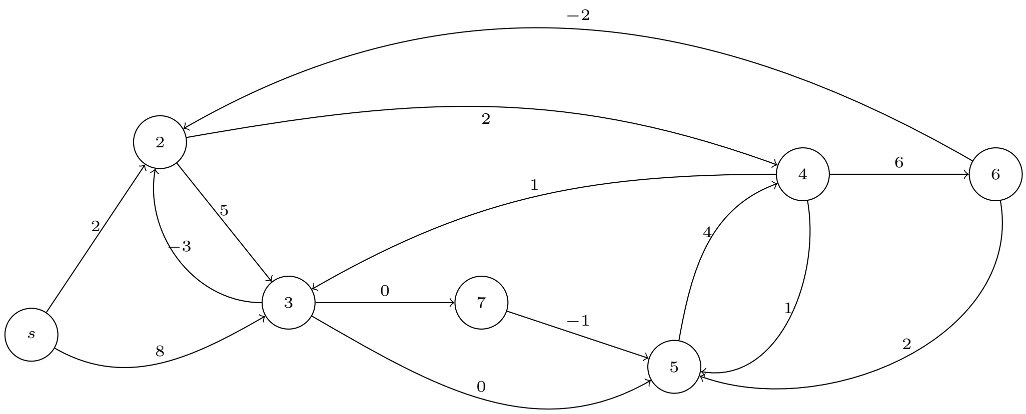
Node i	New final label	Temporarily marked nodes	New labels
Init	-	s	$d(s) = 0, pred(s) = undef$
s	$d(s) = 0, pred(s) = 0$	$\{2, 3\}$	$d(2) = 2, d(3) = 8,$ $pred(2) = pred(3) = s$
2	$d(2) = 2, pred(2) = s$	$\{3, 4\}$	$d(3) = 7, d(4) = 5$ $pred(3) = pred(4) = 2$
4	$d(4) = 5, pred(4) = 2$	$\{3, 5, 6\}$	$d(3) = 6, d(5) = 12, d(6) = 11$ $pred(3) = pred(5) = pred(6) = 4$
3	$d(3) = 6, pred(3) = 4$	$\{5, 6\}$	$d(5) = 6, pred(5) = 3$
5	$d(5) = 6, pred(5) = 3$	$\{6\}$	–
6	$d(6) = 11, pred(6) = 4$	$\emptyset$	–

$\Rightarrow$ Shortest-path-tree:



Exercise 3.3 (Shortest path problems on cyclic graphs with partially negative arc weights)

Given the following graph:

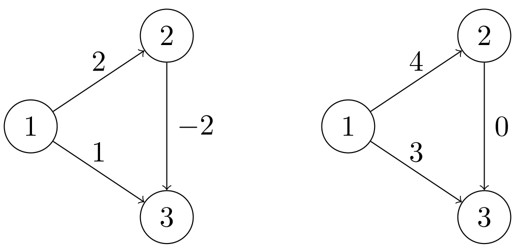


1. To reduce the effort to prepare for the exam, a student decides to learn only the Dijkstra algorithm for solving the shortest path problem. In order to obtain a graph with only non-negative arc weights, she considers simply taking the smallest value among all negative arc weights and adding this amount to all arc weights. In the given graph, for example, she would add 3 to all arc weights and solve the shortest path problem in the resulting graph. Does her approach lead to the desired result? Prove the correctness of the approach or give a counterexample.

2. Solve the shortest path problem in this graph.

Solution:

1. Counterexample:



Shortest path from 1 to 3:

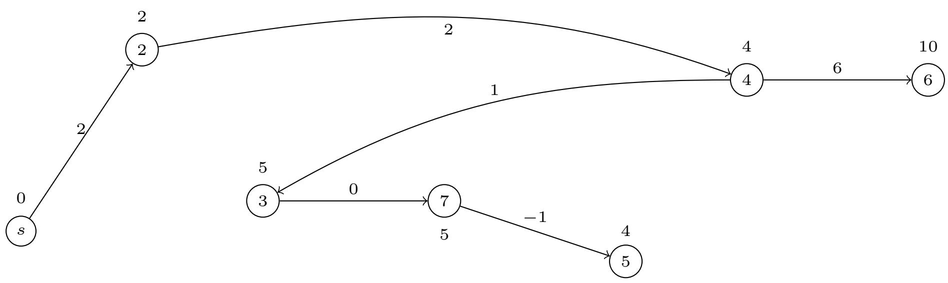
$1 \to 2 \to 3 = 0$	$1 \to 2 \to 3 = 4$
$1 \to 3 = 1$	$1 \to 3 = 3$

2. *FIFO-Algorithm:*

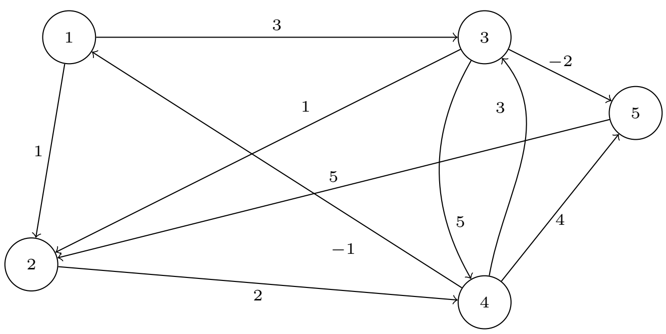
- Prerequisite: no cycle of negative length
- Procedure:
  1. $d(s) = 0$, $pred(s) = undef$, $Q = (s)$, $d(j) = \infty$ for all other nodes
  2. Stop if $Q = \emptyset$
  3. Select first node i from Q, remove i from Q
  4. Check for *all* successors j of i whether $d(j) > d(i) + c_{ij}$. If yes, set $d(j) = d(i) + c_{ij}$, $pred(j) = i$ and if $j \notin Q$, insert j at the end of Q. Goto step 2.

Node i	$\Delta^+(i)$	New labels	*pred*	Queue Q
(Init)	-	$d(s) = 0$	$pred(s) = 0$	(s)
s	$2, 3$	$d(2) = 2, d(3) = 8$	$pred(2) = pred(3) = s$	$(2, 3)$
2	$3, 4$	$d(3) = 7, d(4) = 4$	$pred(3) = pred(4) = 2$	$(3, 4)$
3	$2, 5, 7$	$d(5) = 7, d(7) = 7$	$pred(5) = pred(7) = 3$	$(4, 5, 7)$
4	$3, 5, 6$	$d(3) = 5, d(5) = 5, d(6) = 10$	$pred(3) = pred(5) = pred(6) = 4$	$(5, 7, 3, 6)$
5	4	-	-	$(7, 3, 6)$
7	5	-	-	$(3, 6)$
3	$2, 5, 7$	$d(7) = 5$	$pred(7) = 3$	$(6, 7)$
6	$2, 5$	-	-	(7)
7	5	$d(5) = 4$	$pred(5) = 7$	(5)
5	4	–		$\emptyset$

$\Rightarrow$ Shortest-path-tree:

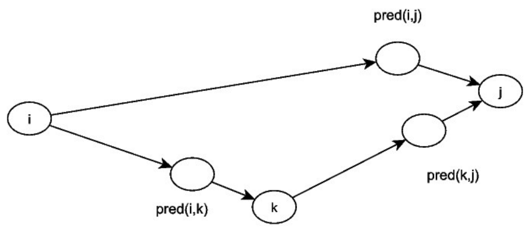


Exercise 3.4 (Shortest path problem between all pairs of nodes)
Find the shortest path between all pairs of nodes in the following digraph.



Solution:
Floyd-Warshall-Algorithm: solve the 'all-pairs' shortest path problem simultaneously

- Prerequisite: no cycles of negative length
- Here, 2-dimensional labels $d(i,j)$, and corresponding $pred(i,j)$
- Procedure:
  1. $V' = V$, $d(i,i) = 0$, $pred(i,i) = undef$, $\forall i \in V$; $d(i,j) = c_{ij}$ and $pred(i,j) = i$, $\forall (i,j) \in A$; $d(i,j) = \infty$ and $pred(i,j) = undef$, $\forall (i,j) \notin A$.
  2. If $V' = \emptyset$, stop.
  3. Select arbitrary node k from V', delete k from V'
  4. Check for all pairs i,j whether: $d(i,j) > d(i,k) + d(k,j)$. If yes, set $d(i,j) = d(i,k) + d(k,j)$, $pred(i,j) = pred(k,j)$, go to step 2.



Matrices describe the solution in stepwise fashion; values modified in red

$d(i,j)$	$pred(i,j)$
Initialization matrix	
$\begin{pmatrix} 0 & 1 & 3 & \infty & \infty \\ \infty & 0 & \infty & 2 & \infty \\ \infty & 1 & 0 & 5 & -2 \\ -1 & \infty & 3 & 0 & 4 \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & - & - \\ - & - & - & 2 & - \\ - & 3 & - & 3 & 3 \\ 4 & - & 4 & - & 4 \\ - & 5 & - & - & - \end{pmatrix}$
$k = 1$	
$\begin{pmatrix} 0 & 1 & 3 & \infty & \infty \\ \infty & 0 & \infty & 2 & \infty \\ \infty & 1 & 0 & 5 & -2 \\ -1 & \textcolor{red}{0} & \textcolor{red}{2} & 0 & 4 \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & - & - \\ - & - & - & 2 & - \\ - & 3 & - & 3 & 3 \\ 4 & \textcolor{red}{1} & \textcolor{red}{1} & - & 4 \\ - & 5 & - & - & - \end{pmatrix}$
$k = 2$	
$\begin{pmatrix} 0 & 1 & 3 & \textcolor{red}{3} & \infty \\ \infty & 0 & \infty & 2 & \infty \\ \infty & 1 & 0 & \textcolor{red}{3} & -2 \\ -1 & 0 & 2 & 0 & 4 \\ \infty & 5 & \infty & \textcolor{red}{7} & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & \textcolor{red}{2} & - \\ - & - & - & 2 & - \\ - & 3 & - & \textcolor{red}{2} & 3 \\ 4 & 1 & 1 & - & 4 \\ - & 5 & - & \textcolor{red}{2} & - \end{pmatrix}$
$k = 3$	
$\begin{pmatrix} 0 & 1 & 3 & 3 & \textcolor{red}{1} \\ \infty & 0 & \infty & 2 & \infty \\ \infty & 1 & 0 & 3 & -2 \\ -1 & 0 & 2 & 0 & \textcolor{red}{0} \\ \infty & 5 & \infty & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & 2 & \textcolor{red}{3} \\ - & - & - & 2 & - \\ - & 3 & - & 2 & 3 \\ 4 & 1 & 1 & - & \textcolor{red}{3} \\ - & 5 & - & 2 & - \end{pmatrix}$
$k = 4$	
$\begin{pmatrix} 0 & 1 & 3 & 3 & 1 \\ \textcolor{red}{1} & 0 & \textcolor{red}{4} & 2 & \textcolor{red}{2} \\ \textcolor{red}{2} & 1 & 0 & 3 & -2 \\ -1 & 0 & 2 & 0 & 0 \\ \textcolor{red}{6} & 5 & \textcolor{red}{9} & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & 2 & 3 \\ \textcolor{red}{4} & - & \textcolor{red}{1} & 2 & \textcolor{red}{3} \\ \textcolor{red}{4} & 3 & - & 2 & 3 \\ 4 & 1 & 1 & - & 3 \\ \textcolor{red}{4} & 5 & \textcolor{red}{1} & 2 & - \end{pmatrix}$
$k = 5$	
$\begin{pmatrix} 0 & 1 & 3 & 3 & 1 \\ 1 & 0 & 4 & 2 & 2 \\ 2 & 1 & 0 & 3 & -2 \\ -1 & 0 & 2 & 0 & 0 \\ 6 & 5 & 9 & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} - & 1 & 1 & 2 & 3 \\ 4 & - & 1 & 2 & 3 \\ 4 & 3 & - & 2 & 3 \\ 4 & 1 & 1 & - & 3 \\ 4 & 5 & 1 & 2 & - \end{pmatrix}$