Arc length, path length

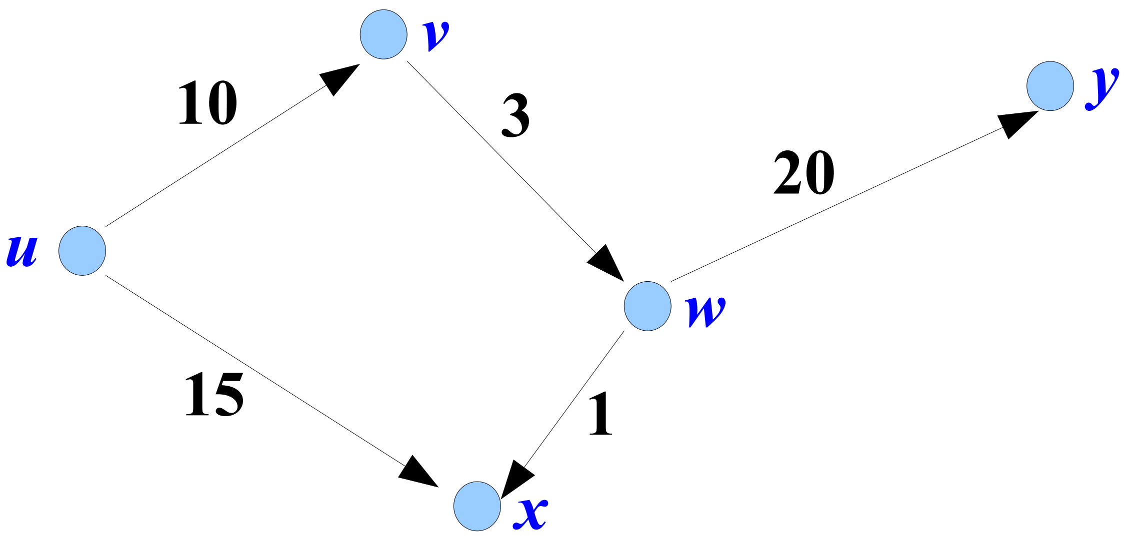
In this section, a graph will always mean a simple digraph.

Let $G=(N, A)$ be a graph and assign a real number $l(a)$ to each arc $a \in A$. This number is called the **length of arc** a.

The **length** $l(p)$ **of path** p in G is defined as the sum of the lengths of all the arcs in p.

Distance, minimal path

Given a graph $G=(N, A)$ and two nodes $u, v \in N$, we define **distance** $d(u, v)$ between *u* and *v* as the minimum length of a path from *u* to *v*. Such **path** is called **minimal** from *u* to *v*. If there is no path from *u* to *v*, put $d(u, v)=\infty$.

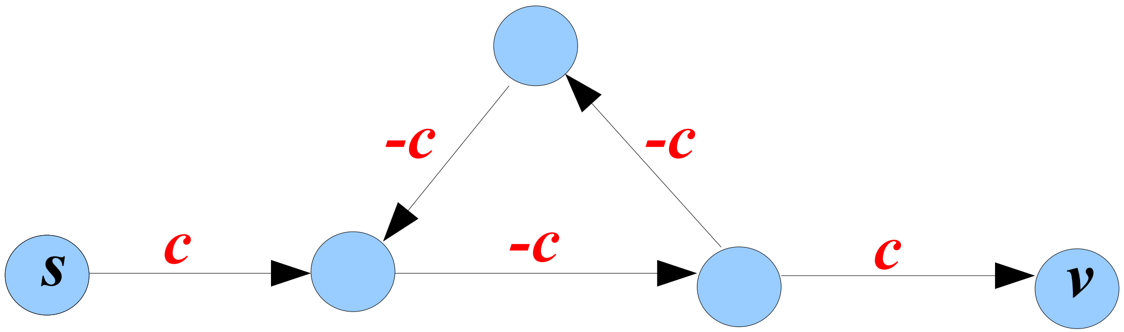


In this graph: $d(u, x)=14, d(u, y)=33$

Note

As the length of an arc may also be negative, the existence of a cycle with a negative length will make the search for a minimal path pointless. As seen in the picture below, there is a walk from *s* to *v* with a length that is less than any given number. To overcome this difficulty, we will first only deal with digraphs that have arcs with positive lengths.

$c > 0$



Consider the following problem:

Let $G=(N, A)$ be a digraph with each arc a assigned a **positive** real number $l(a)$ and let $s \in N$. For every $v \in N$, find the distance $d(s, v)$ and the corresponding minimal path $p(s, v)$.

This problem may be solved by an algorithm devised by
prof. Edsger Wybe Dijkstra*, a Dutch mathematician,*
* 11. 5. 1930 † 6. 8. 2002

Auxiliary concepts

An upper estimate of the distance $d(s,v)$ is a number $D(v)$ such that $D(v) \geq d(s,v)$. For each $v \in N$, $\pi(v)$ will denote the node immediately preceding v in the minimal path from s to v constructed by Dijkstra's algorithm. When such a path has not yet been constructed, put $\pi(v) = \emptyset$.

Next for each $v \in N$, $N(v)$ will denote the set of all nodes to which there is an arc from *v*, formally $N(v) = \{w \in N \mid (v,w) \in A\}$.

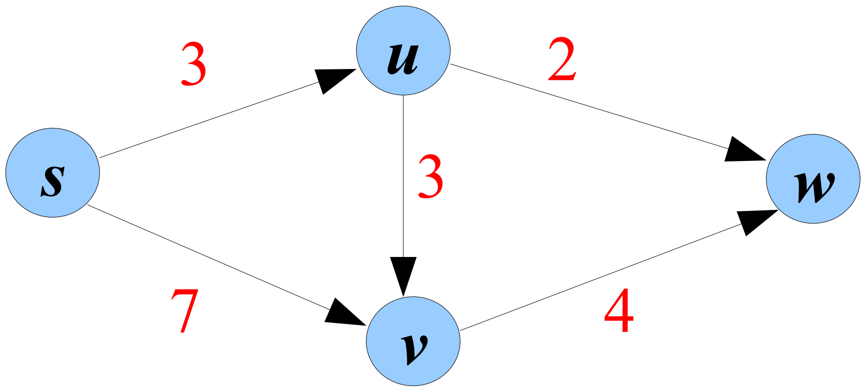
The symbols *S*, $S \subseteq N$ will denote the set of all nodes *w* for which Dijkstra's algorithm has already fixed a minimal path $p(s,v)$ along with $d(s,w)$.

Moreover, $Q = U - S$.

Flow chart

1. **Initialize**: Put $\pi(u) := \emptyset$, $D(s) := 0$, $D(u) := \infty$ if $u \neq s$, $S := \emptyset$ for every $u \in N$
2. **Test for termination**: If $S = N$, go to 5.
3. **Determine fixed node**: In Q find a node v with the least $D(v)$ and move it to S. If $D(u) = \infty$ for all $u \in Q$, go to 5.
4. **Improve upper estimates**: Put $D(w) = D(v) + l((v, w))$ and $\pi(w) = v$ for each $w \in N(v) \cap Q$ such that $D(w) > D(v) + l((v, w))$. Go to 2.
5. **Generate minimal path**:No path exists from s to nodes remaining in Q. For all other nodes put $d(s, v) = D(v)$ and generate path $p(s, v)$ by reversing the path $v \rightarrow \pi(v) \rightarrow \pi(\pi(v)) \rightarrow \pi(\pi(\pi(v))) \rightarrow \cdots \rightarrow s$

Example



Initialize: $S=\{\emptyset\}, Q=\{s,u,v,w\}, \boldsymbol{D(s)=0}, D(u)=D(v)=D(w)=\infty$

Step 1: $S=\{s\}, Q=\{u,v,w\}, D(s)=0, \boldsymbol{D(u)=3}, D(v)=7\ D(w)=\infty$
$\pi(u)=s, \pi(v)=s$

Step 2: $S=\{s,u\}, Q=\{v,w\}, D(s)=0, D(u)=3, D(v)=6, \boldsymbol{D(w)=5}$
$\pi(u)=s, \pi(v)=u, \pi(w)=u$

Step 3: $S=\{s,u,w\}, Q=\{v\}, D(s)=0, D(u)=3, \boldsymbol{D(v)=6}, D(w)=5$
$\pi(u)=s, \pi(v)=u, \pi(w)=u$

Step 4: $S=\{s,u,w,v\}, Q=\emptyset, D(s)=0, D(u)=3, D(v)=6, D(w)=5$
$\pi(u)=s, \pi(v)=u, \pi(w)=u$

$p(s,u)=s\rightarrow u, p(s,v)=s\rightarrow u\rightarrow v, p(s,w)=s\rightarrow u\rightarrow w$

Theorem

For every $v \in N$, Dijkstra's algorithm will find a minimal path $p(s, v)$ and the distance $d(s, v)$.

Proof:

We will prove that, at any time during the algorithm's procedure, we have for every $v \in S$,: $D(v) = d(s, v)$ and the corresponding path from *s* to *v* is only built from nodes in *S*. Now this is certainly true if $S = \emptyset$. Suppose that, at a certain point, this is true for every node in *S*. Thus, immediately before moving a new node $v \in Q$ to *S*, the situation is as follows:

(1) $D(w) = d(s, w)$ for $w \in S$,

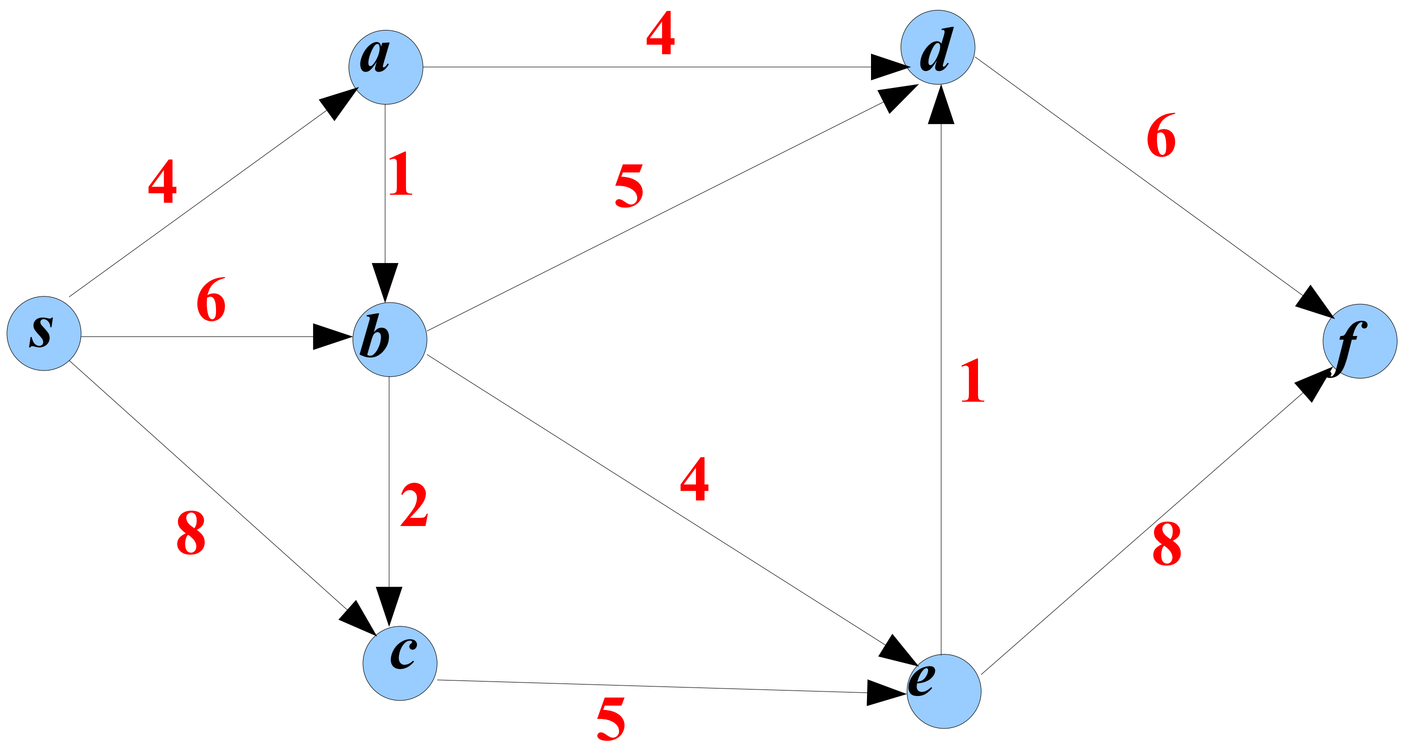
(2) if $w \in S$, then $u \in S$ for every node *u* included in the path $p(s, w)$.

(3) for every $w \in Q$, $D(w)$ is the length of a minimal path from s to w such that $u \in S$ for its every node u, $u \neq w$,

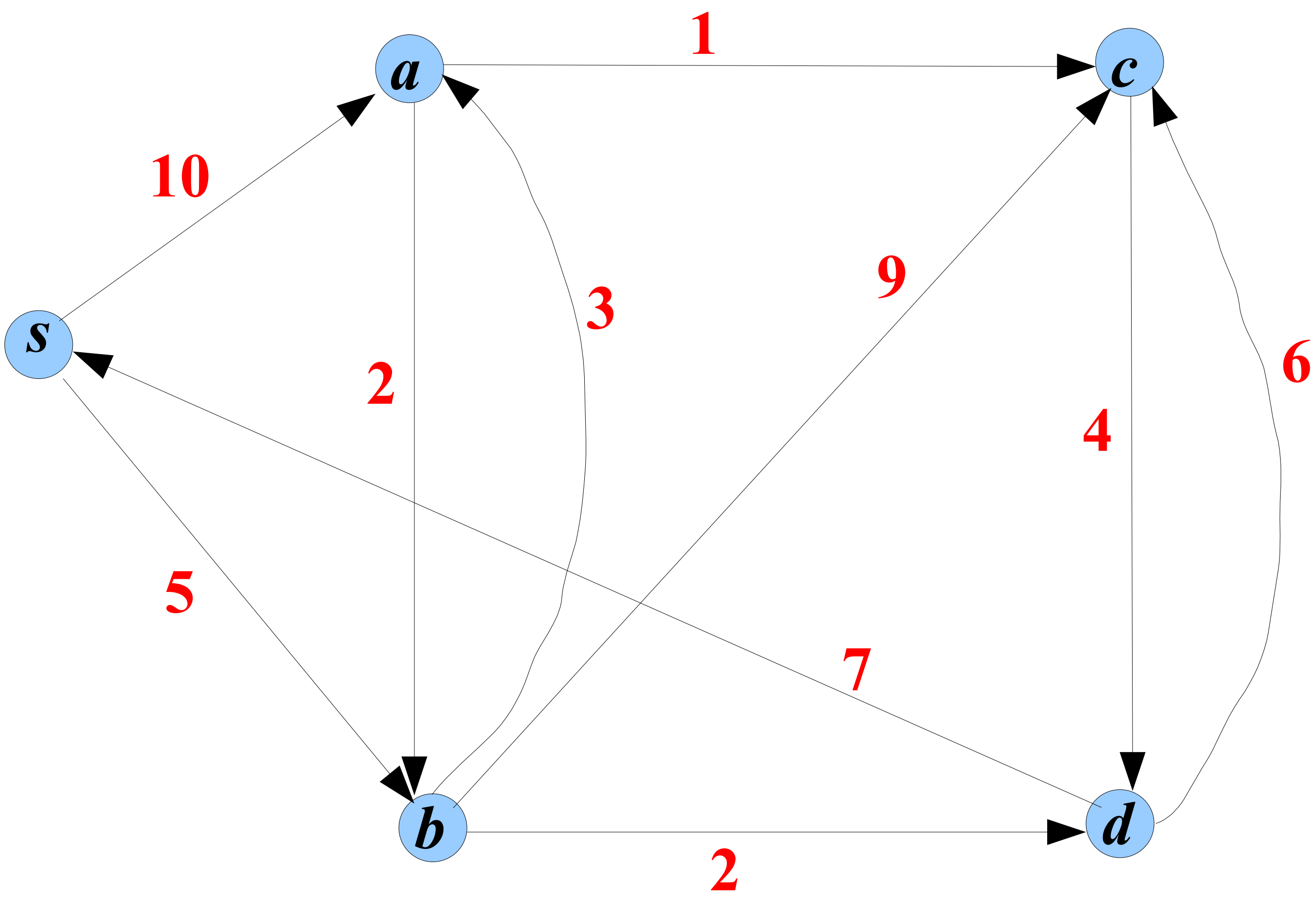
(4) $D(v) \leq D(u)$ for $u \in Q$

Suppose now that there is a path p' from s to v such that $l(p') \leq D(v)$ and at least one node z in p' such that $z \in Q$. Without loss of generality, we can think of z as the first such node from s in the path p'. We have $D(z) < D(v)$ since the length of every arc is positive. However, this contradicts (4). This means that, when node v is moved to S, properties (1) to (4) remain in force.

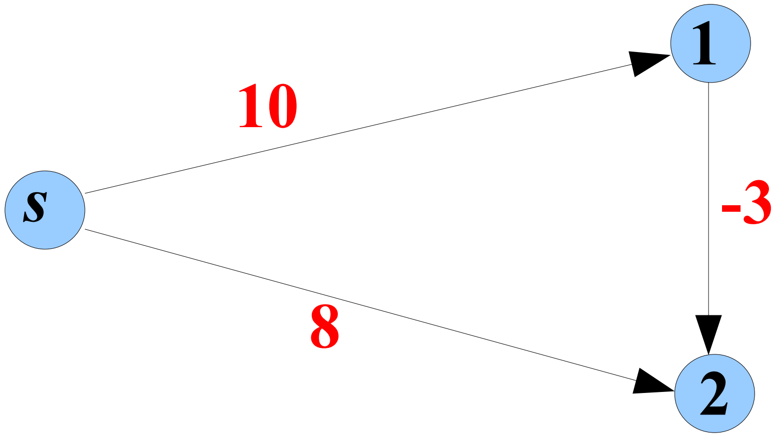
Exercise 1



Exercise 2



Problem that cannot be solved by Dijkstra's algorithm



It can be verified easily that, after Dijkstra's algorithm terminates, we have:

$D(s)= 0$, $D(1) = 10$, $D(2) = 8$.

However, it is evident that the minimal path from node 2 has a value of 7.

This is due to the negative length of arc (1,2) even if there are no cycles with a negative length.

Floyd-Warshall's algorithm

The above example shows that Dijkstra's algorithm does not work for graphs containing negative lengths. For such graphs, provided that they do not contain-ing cycles with negative lengths, Floyd-Warshall's algorithm may be an alternat-ive. If the lengths of arcs are given, this algorithm will find a minimal path from each node to each node and, if such a path does not exist due to a negative cycle, it will be detected.

Consider graph $G=(U,H)$ with n nodes and the lengths given as entries in the following matrix

$$A=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

Next we will also use a matrix

$$P=\begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

with initial values $p_{ij}=j$

The algorithm has always n iterations:

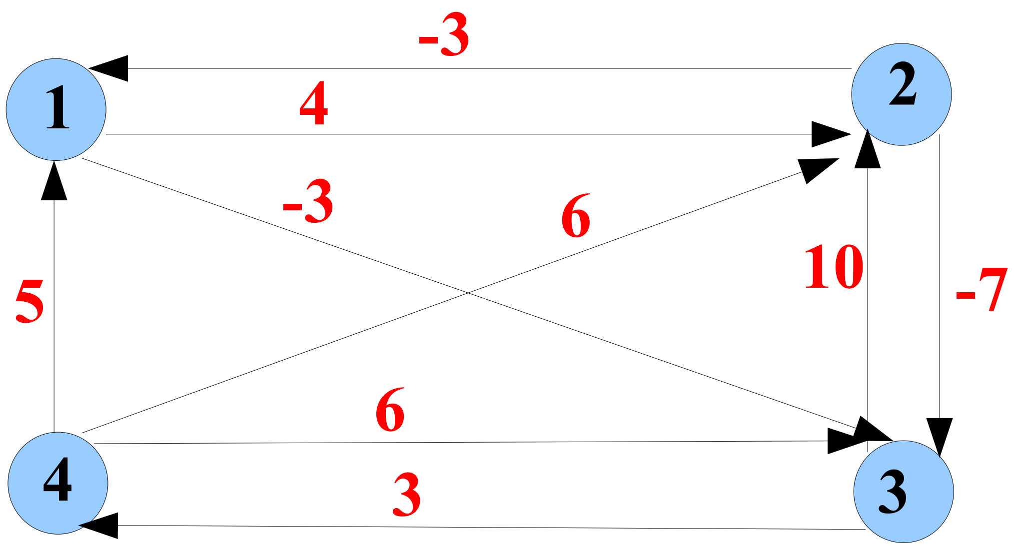
We start off with matrices $A^0 = A, P^0 = P$ and, in iteration i, create matrices A^i, P^i from matrices A^{i-1}, P^{i-1} respectively. Thus, in the last iteration, we will create matrices A^n, P^n. The entries of matrices $A^j, P^j, j = 1,2,\ldots,n$ are calculated as follows:

$$a_{ik}^j = a_{ik}^{j-1}, p_{ik}^j = p_{ik}^{j-1} \text{ if } a_{ik}^{j-1} \le a_{ij}^{j-1} + a_{jk}^{j-1}$$

$$a_{ik}^j = a_{ij}^{j-1} + a_{jk}^{j-1}, p_{ik}^j = p_{ij}^{j-1} \text{ if } a_{ik}^{j-1} > a_{ij}^{j-1} + a_{jk}^{j-1}$$

It may be proved by induction that, after the algorithm terminates, entry a_{ij}^n has the value of the distance from node i to node j. It can also be verified that $p_{ij}^n = k$ if (i, k) is the first arc in a minimal path from node i to node j, which can be used to determine such a path.

Example



$$A=\begin{pmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ \infty & 10 & 0 & 3 \\ 5 & 6 & 6 & 0 \end{pmatrix} \qquad P=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Iteration 0

$$A^{(0)}=\begin{pmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ \infty & 10 & 0 & 3 \\ 5 & 6 & 6 & 0 \end{pmatrix} \quad P^{(0)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Iteration 1

$$A^{(1)}=\begin{pmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ \infty & 10 & 0 & 3 \\ 5 & 6 & 2 & 0 \end{pmatrix} \quad P^{(1)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 4 \end{pmatrix}$$

Iteration 2

$$A^{(2)}=\begin{pmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ 7 & 10 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{pmatrix} \quad P^{(2)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix}$$

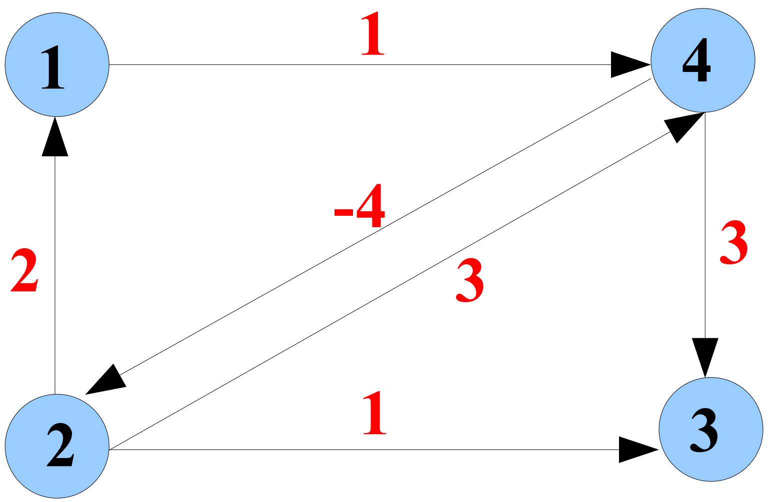
Iteration 3

$$A^{(3)}=\begin{pmatrix} 0 & 4 & -3 & 0 \\ -3 & 0 & -7 & -4 \\ 7 & 10 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{pmatrix} \quad P^{(3)}=\begin{pmatrix} 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 2 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix}$$

Iteration 4

$$A^{(4)}=\begin{pmatrix} 0 & 4 & -3 & 0 \\ -3 & 0 & -7 & -4 \\ 6 & 9 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{pmatrix} \quad P^{(4)}=\begin{pmatrix} 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 4 & 4 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{pmatrix}$$

Example



$$A = \begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Iteration 0

$$A^{(0)}=\begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix} \quad P^{(0)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Iteration 1

$$A^{(1)}=\begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix} \quad P^{(1)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Iteration 2

$$A^{(2)}=\begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ -2 & -4 & -3 & \mathbf{-1} \end{pmatrix} \quad P^{(2)}=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 2 \end{pmatrix}$$

Here the diagonal entry (4,4) is negative, which indicates the existence of a cycle with a negative length and thus the non-existence of a minimal path.