Dijkstra-Algorithmus:

1. Man setze $l(v_0) = 0$, $l(v) := \infty$ für alle $v \in V \setminus \{v_0\}$, $U = \{v_0\}$ und $u = v_0$.

2. Für alle $v \in V \setminus U$ mit $(u, v) \in E$, die $l(v) > l(u) + w(u, v)$ erfüllen, setze man $p(v) := u$ und
$$l(v) := l(u) + w(u, v).$$

3. Man bestimme $m = \min_{v \in V \setminus U} l(v)$, wähle eine Knoten $z \in V \setminus U$ mit $l(z) = m$ und setze $U := U \cup \{z\}$ und $u := z$.

4. Ist $U = V$ oder $l(v) = \infty$ für alle $v \in V \setminus U$, so wird der Algorithmus beendet. Andernfalls gehe zu Schritt 2.

Floyd-Warshall Algorithm

$V = \{v_1, v_2, \ldots, v_n\}, \quad W = (w_{ij})_{1 \le i,j \le n}, \quad w_{ij} = w(v_i, v_j)$

(1) $\ell_{ij} := w_{ij}$

(2) for i=1 to n do
```
    for j=1 to n do
      for k=1 to n do
        ℓ_jk := min(ℓ_jk, ℓ_ji + ℓ_ik)
      end
      if  ℓ_jk < 0 then STOP (Cycle of negative length!)
    end
end
```

Moore's Algorithm

1. Setze $l(v_0) := 0$, $a(v_0) := 0$, $p(v_0) := *$ und für alle Knoten $v \in V(G) \setminus \{v_0\}$ $l(v) := \infty$, $a(v) := 1$, $p(v) := *$. Weiters setze für alle Kanten $e = (v, w)$, $v \neq w$, die nicht in G liegen $w(e) := \infty$, für alle Schlingen $w(v, v) := 0$ und schließlich $ANZ := 0$ und $IND := 0$.
2.
```
for j := 0 to n do
  if a(v_j) = ANZ then
    for k = 1 to n
      if l(v_j) + w(v_j, v_k) < l(v_k) then
        IND := 1
        l(v_k) := l(v_j) + w(v_j, v_k)
        a(v_k) := a(v_j) + 1
        p(v_k) := p(v_j)
      endif;
    end;
  endif;
end;
```
3. Ist $IND = 0 \rightarrow$ **ENDE.**
   Ist $IND = 1$, dann führe folgendes aus:
   $ANZ := ANZ + 1$
   if $ANZ \geq n$ then $\rightarrow$ **ENDE**
   (Es gibt einen Zyklus negativer Länge.)
   Gehe zu 2.

Algorithm of Ford and Fulkerson

```
(1) for  e ∈ E do: φ(e) := 0: end;
(2) p(s) := +s, δ(s) := ∞, V_1 := {s}, V_2 := V \ {s}
(3) for  x ∈ V_1 do:
    (a) for y ∈ Γ^+(x) ∩ V_2 do:
            if φ(⟨x,y⟩) < w(⟨x,y⟩) then
               p(y) := +x;
               δ(y) := min(δ(x), w(⟨x,y⟩) − φ(⟨x,y⟩));
               V_1 := V_1 ∪ {y}; V_2 := V_2 \ {y};
            end;
    (b) for y ∈ Γ^−(x) ∩ V_2 do:
            if φ(⟨x,y⟩) > 0 then
               p(y) := −x;
               δ(y) := min(δ(x), φ(⟨x,y⟩));
               V_1 := V_1 ∪ {y}; V_2 := V_2 \ {y};
            end;
(4) if  V_1 increased in step 3. or  t ∈ V_2 then goto 3.
(5) if  t ∈ V_2 then STOP
    else
        x := t;
        while x ≠ s do
                if p(x) = +z then
                   φ(⟨z,x⟩) := φ(⟨z,x⟩) + δ(t);
                   x := z;
                end;
                if p(x) = −z then
                   φ(⟨z,x⟩) := φ(⟨z,x⟩) − δ(t);
                   x := z;
                end;
        end;
(6) goto 2
```