Συντομότερα Μονοπάτια

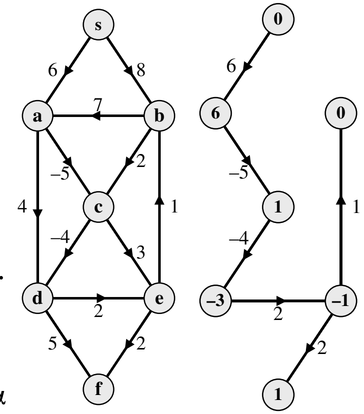
Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

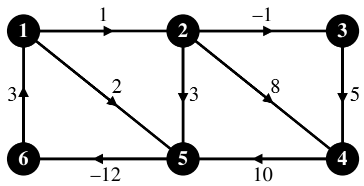
Συντομότερα Μονοπάτια (ΣΜ)

- Κατευθυνόμενο γράφημα $G(V, E, w)$ με μήκη στις ακμές $w : E \mapsto \mathbb{R}$.
- Μήκος μονοπατιού $p = (v_0, v_1, \ldots, v_k)$:
  $w(p) = \sum_{i=0}^{k} w(v_{i-1}, v_i)$.
- Απόσταση $d(u, v)$ = μήκος συντομότερου μονοπατιού $u - v$.
  $\nexists$ μονοπάτι $u - v \Rightarrow d(u, v) = \infty$.
- **Συντομότερο $u - v$ Μονοπάτι** μήκους $d(u, v)$.
- Ζητούμενο: Συντομότερα μονοπάτια από αρχική κορυφή s προς όλες τις κορυφές.
- Υπολογισμός ΣΜ(s): Θεμελιώδες πρόβλημα συνδυαστικής βελτιστοποίησης!



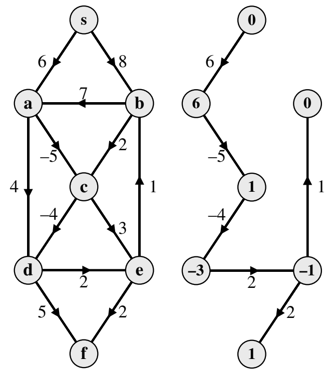
Κύκλοι Αρνητικού Μήκους

- **Διαδρομή**: ακολουθία κορυφών όπου διαδοχικές συνδέονται με ακμή.
- **Περίπατος**: διαδρομή χωρίς επαναλαμβανόμενες ακμές.
- **Μονοπάτι**: διαδρομή χωρίς επαναλαμβανόμενες κορυφές.
- Συντομότερη διαδρομή είναι μονοπάτι εκτός αν ...
  υπάρχει **κύκλος αρνητικού μήκους**.
  - Αποστάσεις **δεν ορίζονται** γιατί συνολικό μήκος διαδρομής **μειώνεται όσο περισσότερο** χρησιμοποιεί έναν κύκλο αρνητικού μήκους.
  - Κύκλος αρνητικού μήκους σε $u - v$ μονοπάτι $\Rightarrow d(u, v) = -\infty$.



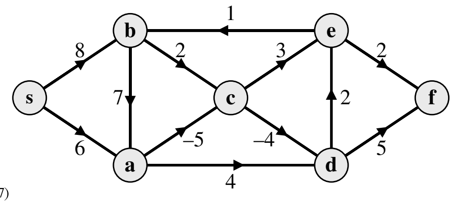
Ιδιότητες Συντομότερων Μονοπατιών

- $p = (v_0, v_1, \ldots, v_k)$ συντομότερο μονοπάτι.
  Κάθε τμήμα του p είναι συντομότερο $v_i - v_j$ μονοπάτι.
  Ιδιότητα Βέλτιστων Επιμέρους Λύσεων.
- **Δέντρο Συντομότερων Μονοπατιών** από s.
  ΔΣΜ $\rightarrow$ πίνακας προγόνων $p[1 \ldots n]$.
- **Ερώτηση**: ΔΣΜ = ΕΕΔ ; **ΟΧΙ**
- $\forall (u, v) \in E$, $d(s, u) \leq d(s, v) + w(v, u)$
  Τριγωνική Ανισότητα!
- Τι συμβαίνει με συντομότερα μονοπάτια αν πολλαπλασιάσουμε μήκη με θετικό αριθμό;
- Τι συμβαίνει αν προσθέσουμε στα μήκη έναν θετικό αριθμό;



Υπολογισμός Συντομότερων Μονοπατιών

- "Απαισιόδοξη" εκτίμηση $\ell[v]$ για $d(s,v)$.
  Αρχικά $\ell[s] = 0$ και $\ell[v] = \infty$ $\forall v \in V \setminus \{s\}$.
- Αλγόριθμος **εξετάζει ακμές** (v,u) και **αναπροσαρμόζει** $\ell[u]$.
  Αν $\ell[u] > \ell[v] + w(v,u)$ τότε $\ell[u] \leftarrow \ell[v] + w(v,u)$.
  $p[u] \leftarrow v;$
- $\ell[v] =$ μήκος συντομότερου **γνωστού** $s - v$ μονοπατιού.
  Επαγωγικά: Ισχύει **πριν** τελευταία εξέταση ακμής $(v,u) \Rightarrow$
  Ισχύει **μετά** γιατί $\ell'[u] = \min\{\ell[u], \ell[v] + w(u,v)\}$.
- Πάντα $\ell[v] \geq d(s,v)$. $\nexists\, s - v$ μονοπάτι $\Rightarrow \ell[v] = \infty$ για πάντα.
- **Συστηματικό τρόπο εξέτασης ακμών** και **κριτήριο τερματισμού**.

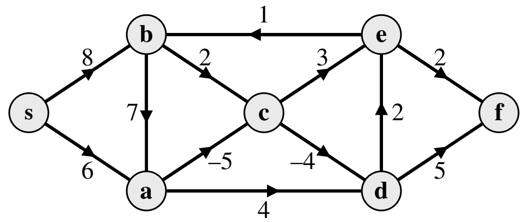


Αλγόριθμος Bellman-Ford : Δυναμικός Προγραμματισμός

- $L(i,u) =$ μήκος συντομότερου $s - u$ μονοπατιού με i ή λιγότερες ακμές.
- Αρχικά, $L(0,s) = 0$ και $L(0,u) = \infty$ $\forall u \in V \setminus \{s\}$.

$$L(i+1,u) = \min\{L(i,u), \min_{v:(v,u)\in E}\{L(i,v) + w(v,u)\}\}$$

- (Απλό) μονοπάτι $\leq n-1$ ακμές $\Rightarrow L(n-1,u) = d(s,u)$.
  $L(n,u) < L(n-1,u) \Leftrightarrow$ κύκλος αρνητικού μήκους.
- Υπολογισμός $L(n-1,u)$ $\forall u \in V$ με **δυναμικό προγραμματισμό**.

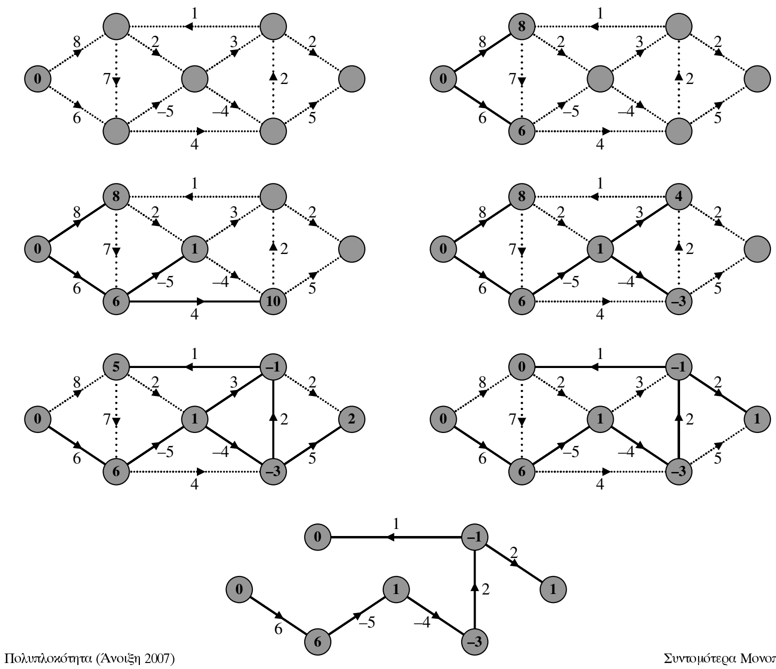


Αλγόριθμος Bellman-Ford : Υλοποίηση

- "Απαισιόδοξη" εκτίμηση $\ell[u]$:
  Τέλος φάσης i, $\ell[u] \leq L(i,u)$.
- Σε κάθε φάση $i = 1, \ldots, n-1$,
  κάθε ακμή εξετάζεται μία φορά.
- Επιπλέον φάση για κύκλο αρνητικού μήκους.
- Χρόνος Εκτέλεσης: $\Theta(nm)$.

Bellman-Ford$(G(V,E,w), s)$
  for all $u \in V$ **do**
    $\ell[u] \leftarrow \infty$; $p[u] \leftarrow$ NIL;
  $\ell[s] \leftarrow 0$;
  for $i \leftarrow 1$ **to** $n-1$ **do**
    for all $(v,u) \in E$ **do**
      if $\ell[u] > \ell[v] + w(v,u)$ **then**
        $\ell[u] \leftarrow \ell[v] + w(v,u)$;
        $p[u] \leftarrow v$;
  for all $(v,u) \in E$ **do**
    if $\ell[u] > \ell[v] + w(v,u)$ **then**
      return(ΚΥΚΛΟΣ ΑΡΝΗΤ);

Αλγόριθμος Bellman-Ford : Παράδειγμα

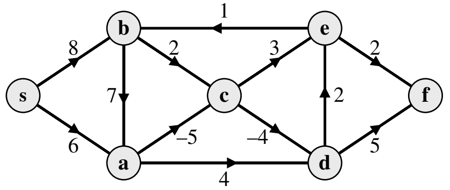


Αλγόριθμος Bellman-Ford : Ορθότητα

- Αν **δεν υπάρχει** κύκλος αρνητικού μήκους, $\ell[u] = d(s,u)$ στο τέλος.
  - Συντομότερο $s-u$ μονοπάτι $(s = v_0, v_1, \ldots, v_k = u)$ με $k \le n-1$ ακμές.
  - Επαγωγική υπόθεση: Τέλος φάσης $i-1$, $\ell[v_{i-1}] = d(s, v_{i-1})$.
  - Τέλος φάσης i: Εξέταση ακμής (v_{i-1}, v_i) και $\ell[v_i] = d(s, v_i)$ γιατί:

$$d(s,v_i) \le \ell[v_i] \le \ell[v_{i-1}] + w(v_{i-1},v_i) = d(s,v_{i-1}) + w(v_{i-1},v_i) = d(s,v_i)$$

  - Τέλος φάσης $n-1$: $\ell[u] = d(s,u)$ για όλες τις κορυφές u. $\ell[u]$ δεν μικραίνει άλλο γιατί πάντα $\ell[u] \ge d(s,u)$.
  - Αλγόριθμος δεν επιστρέφει ένδειξη για κύκλο αρνητικού μήκους.

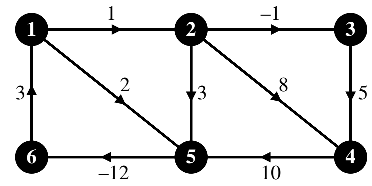


Αλγόριθμος Bellman-Ford : Ορθότητα

- Αν **υπάρχει** κύκλος αρνητικού μήκους, ένδειξη στο τέλος.
  - Κύκλος αρνητικού μήκους $(v_0, v_1, \ldots, v_k)$, $v_0 = v_k$, προσπελάσιμος από $s \Rightarrow \ell[v_i]$ πεπερασμένη τέλος φάσης $n-1$.
  - Έστω όχι ένδειξη $\Rightarrow$ φάση n: $\ell[v_i] \le \ell[v_{i-1}] + w(v_{i-1}, v_i)$ για κάθε v_i.
  - Άθροιση κατά μέλη:

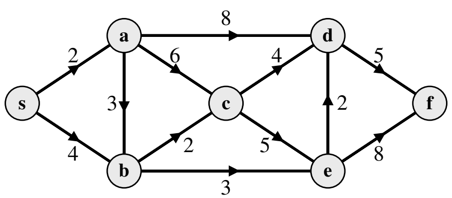
$$\sum_{i=1}^{k} \ell[v_i] \le \sum_{i=1}^{k} \ell[v_{i-1}] + \sum_{i=1}^{k} w(v_{i-1}, v_i) \Rightarrow \sum_{i=1}^{k} w(v_{i-1}, v_i) \ge 0$$

  - Άτοπο! Άρα **ένδειξη για κύκλο αρνητικού μήκους** στο τέλος.



Αλγόριθμος Dijkstra : Απληστία

- Γρηγορότερα όταν **δεν υπάρχουν αρνητικά μήκη**. Γενίκευση της Αναζήτησης Πρώτα σε Πλάτος.
- Μία-μία κορυφές στο ΔΣΜ σε **αύξουσα σειρά απόστασης** από s.
  - Αρχικά κενό ΔΣΜ, $\ell[s] \leftarrow 0$, και $\ell[u] \leftarrow \infty$ για κάθε κορυφή $u \ne s$.
  - Κορυφή $u \notin S$ με ελάχιστο $\ell[u]$ εντάσσεται στο ΔΣΜ.
  - Για κάθε ακμή (u,v), $\ell[v] \leftarrow \min\{\ell[v], \ell[u] + w(u,v)\}$.
- Όταν u εντάσσεται στο ΔΣΜ, $\ell[u] = d(s,u)$. **Θετικά μήκη** $\Rightarrow$ κορυφές σε μεγαλύτερη απόσταση δεν επηρεάζουν $\ell[u]$!



Αλγόριθμος Dijkstra : Υλοποίηση

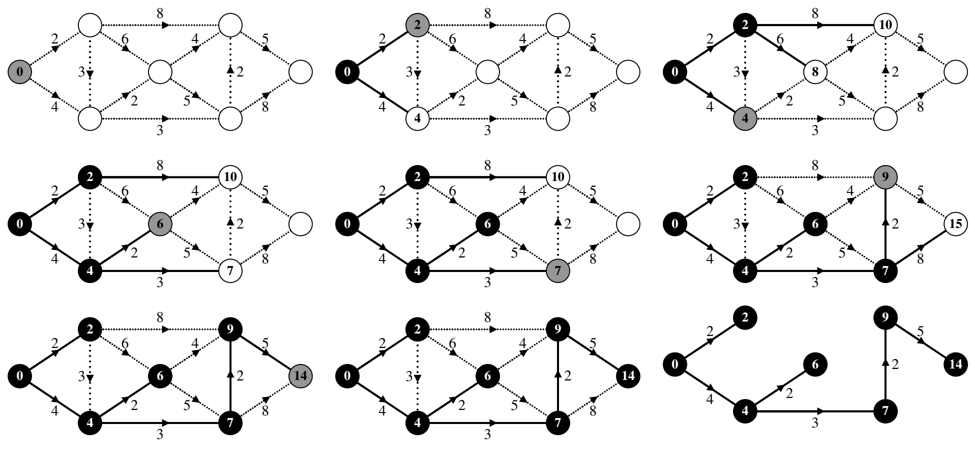
- **Υλοποίηση:** Ελάχιστο $\ell[u] \Rightarrow$ Ουρά Προτεραιότητας.

  Binary Heap: $\Theta(m \log n)$.

  Fibonacci Heap: $\Theta(m + n \log n)$.

```
Dijkstra(G(V,E,w), s)
  for all u ∈ V do
    ℓ[u] ← ∞;  p[u] ← NIL;
  ℓ[s] ← 0;  S ← ∅;
  while |S| < n do
    u ∉ S : ℓ[u] = min_{v∉S}{ℓ[v]};
    S ← S ∪ {u};
    for all v ∈ AdjList[u] do
      if ℓ[v] > ℓ[u] + w(u,v) then
        ℓ[v] ← ℓ[u] + w(u,v);
        p[v] ← u;
```

Αλγόριθμος Dijkstra : Παράδειγμα



Κάτι Μου Θυμίζει ...;!

Dijkstra$(G(V,E,w),s)$
- **for all** $u \in V$ **do**
  - $\ell[u] \leftarrow \infty$; $p[u] \leftarrow$ NIL;
- $\ell[s] \leftarrow 0$; $S \leftarrow \emptyset$;
- **while** $|S| < n$ **do**
  - $u \notin S : \ell[u] = \min_{v \notin S}\{\ell[v]\}$;
  - $S \leftarrow S \cup \{u\}$;
  - **for all** $v \in$ AdjList$[u]$ **do**
    - **if** $\ell[v] > \ell[u] + w(u,v)$ **then**
      - $\ell[v] \leftarrow \ell[u] + w(u,v)$;
      - $p[v] \leftarrow u$;

Prim$(G(V,E,w),s)$
- **for all** $u \in V$ **do**
  - $c[u] \leftarrow \infty$; $p[u] \leftarrow$ NIL;
- $c[s] \leftarrow 0$; $S \leftarrow \emptyset$; $\Delta \leftarrow \emptyset$;
- **while** $|S| < n$ **do**
  - $u \notin S : c[u] = \min_{v \notin S}\{c[v]\}$;
  - $S \leftarrow S \cup \{u\}$;
  - **for all** $v \in$ AdjList$[u]$ **do**
    - **if** $v \notin S$ **and** $w(u,v) < c[v]$ **then**
      - $c[v] \leftarrow w(u,v)$;
      - $p[v] \leftarrow u$;
  - **if** $p[u] \neq$ NIL **then**
    - $\Delta \leftarrow \Delta \cup \{u, p[u]\}$;

Αλγόριθμος Dijkstra : Ορθότητα

Θ.δ.ο. όταν κορυφή u εντάσσεται στο ΔΣΜ, $\ell[u] = d(s,u)$.

Επαγωγή: ΄Εστω $\ell[v] = d(s,v)$ για όλες τις κορυφές πριν τη u.

Κορυφή u έχει **ελάχιστο** $\ell[u]$. ΄Εστω ότι $\ell[u] > d(s,u)$.

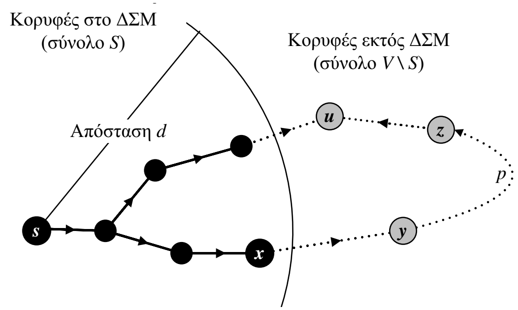
p συντομότερο $s-u$ μονοπάτι μήκους $d(s,u) < \ell[u]$ και z πριν u στο p:

$$d(s,u) = d(s,z) + w(z,u) < \ell[u]$$
$$\Rightarrow d(s,z) < \ell[u]$$
$$\Rightarrow z \notin S$$

x τελευταία p στο S, y επόμενη.

$$\begin{aligned} \ell[y] &\leq \ell[x] + w(x,y) \\ &= d(s,x) + w(x,y) \\ &\leq d(s,z) < \ell[u] \end{aligned}$$

$\Rightarrow \ell[y] < \ell[u]$, **άτοπο!**



Dijkstra vs Bellman-Ford

- Dijkstra **ταχύτερος κατά** n αλλά **όχι αρνητικά μήκη**.
- Αποστάσεις δεν μειώνονται κατά μήκος μονοπατιού.
- Bellman-Ford ακμές **αρνητικού μήκους**.
- Αποστάσεις μπορεί να μειώνονται κατά μήκος μονοπατιού.
  Τελευταία κορυφή μπορεί να έχει μικρότερη απόσταση από προηγούμενες.

Ασκήσεις

- Αρνητικά μήκη → Προσθέτω μεγάλο αριθμό → **Θετικά μήκη** → Αλγόριθμος Dijkstra;
- Ν.δ.ο. Αναζήτηση Πρώτα σε Πλάτος υπολογίζει ένα ΔΣΜ όταν ακμές **μοναδιαίου μήκους**.
- **Πρόβλημα Στενωπού**: Κόστος μονοπατιού $p = \max_{e \in p}\{w(e)\}$.
  Υπολογισμός $s - t$ μονοπατιού με ελάχιστο κόστος.
  Να τροποποιήσετε τον αλγόριθμο Dijkstra για να λύνει Πρόβλημα Στενωπού (και για αρνητικά μήκη).

$$\forall (u,v) \in E, \;\; \ell[v] \leftarrow \min\{\ell[v], \max\{\ell[u], w(u,v)\}\}$$

Συντομότερα Μονοπάτια για Όλα τα Ζεύγη Κορυφών

- Απόσταση $d(u,v)$ και συντομότερο $u - v$ μονοπάτι $\forall (u,v) \in V \times V$.
- Εκτέλεση για κάθε διαφορετική αρχική κορυφή $s \in V$:
  - **Αρνητικά βάρη** → Bellman-Ford: $\Theta(n^2 m)$.
  - **Μη-αρνητικά βάρη** → Dijkstra: $\Theta(nm + n^2 \log n)$.
- **Αρνητικά βάρη** → Floyd-Warshall: $\Theta(n^3)$.
- **Αναπαράσταση λύσης**:
  - **Αποστάσεις**: Τετραγωνικός πίνακας $D[1 \ldots n][1 \ldots n]$.
  - **Συντομότερα Μονοπάτια**: n ΔΣΜ (ένα για κάθε αρχική κορυφή).
    n πίνακες προγόνων $p[1 \ldots n]$ → πίνακας $P[1 \ldots n][1 \ldots n]$.
    Γραμμή $P[i]$ πίνακας προγόνων ΔΣΜ(v_i).

Αλγόριθμος Floyd-Warshall : Δυναμικός Προγραμματισμός

- $G(V, E, w)$. Θεωρούμε $V = \{v_1, v_2, \ldots, v_n\}$.
- Αναπαράσταση γραφήματος με **βεβαρυμένο πίνακα γειτνίασης**:

$$w(v_i, v_j) = \begin{cases} 0 & \text{αν } v_i = v_j \\ w(v_i, v_j) & \text{αν } v_i \neq v_j \text{ και ακμή } (v_i, v_j) \in E \\ \infty & \text{αν } v_i \neq v_j \text{ και ακμή } (v_i, v_j) \notin E \end{cases}$$

- Απόσταση $d(v_i, v_j)$ από $d(v_i, v_k), d(v_k, v_j)$ για κάθε $v_k \in V \setminus \{v_i, v_j\}$:

$$d(v_i, v_j) = \min\{w(v_i, v_j), \min_{v_k \in V \setminus \{v_i, v_j\}} \{d(v_i, v_k) + d(v_k, v_j)\}\}$$

- Μοιάζει με **φαύλο κύκλο**: $d(v_i, v_k) \to d(v_i, v_j)$ και $d(v_i, v_j) \to d(v_i, v_k)$.
- Υπολογισμός όλων με **συστηματικό bottom-up τρόπο**:
  Δυναμικός Προγραμματισμός!

Αλγόριθμος Floyd-Warshall : Δυναμικός Προγραμματισμός

- $D_k[v_i, v_j]$ = μήκος συντομότερου $v_i - v_j$ μονοπατιού με
  ενδιάμεσες κορυφές μόνο από $V_k = \{v_1, \ldots, v_k\}$.
- Αρχικά $D_0[v_i, v_j] = w(v_i, v_j)$ γιατί $V_0 = \emptyset$.
- Έστω ότι γνωρίζω $D_{k-1}[v_i, v_j]$ για όλες τις κορυφές v_i, v_j.
  $D_k[v_i, v_j]$ διέρχεται από v_k καμία ή μία φορά (**μονοπάτι**).

$$D_k[v_i, v_j] = \min\{D_{k-1}[v_i, v_j], D_{k-1}[v_i, v_k] + D_{k-1}[v_k, v_j]\}$$

- Αναδρομική σχέση για $D_0, D_1, \ldots, D_n$ με **δυναμικό προγραμματισμό**:

$$D_k[v_i, v_j] = \begin{cases} w(v_i, v_j) & k = 0 \\ \min\{D_{k-1}[v_i, v_j], D_{k-1}[v_i, v_k] + D_{k-1}[v_k, v_j]\} & k = 1, \ldots, n \end{cases}$$

Αλγόριθμος Floyd-Warshall : Υλοποίηση

- **Τυπικός** Δυναμικός Προγραμματισμός: $\begin{cases} \textbf{Χρόνος:} & \Theta(n^3) \\ \textbf{Μνήμη:} & \Theta(n^2) \end{cases}$

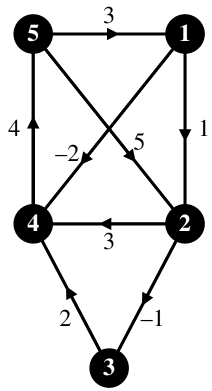
Floyd-Warshall($G(V, E, w)$)
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
      if $(v_i, v_j) \in E$ **then** $D_0[i, j] \leftarrow w(v_i, v_j)$;
      else $D_0[i, j] \leftarrow \infty$;
    $D_0[i, i] \leftarrow 0$;
  for $k \leftarrow 1$ **to** n **do**
    for $i \leftarrow 1$ **to** n **do**
      for $j \leftarrow 1$ **to** n **do**
        if $D_{k-1}[i, j] > D_{k-1}[i, k] + D_{k-1}[k, j]$ **then**
          $D_k[i, j] \leftarrow D_{k-1}[i, k] + D_{k-1}[k, j]$;
        else $D_k[i, j] \leftarrow D_{k-1}[i, j]$;

Αλγόριθμος Floyd-Warshall : Παράδειγμα

$$D_0 = \begin{pmatrix} 0 & 1 & \infty & -2 & \infty \\ \infty & 0 & -1 & 3 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 4 \\ 3 & 5 & \infty & \infty & 0 \end{pmatrix} \quad D_1 = \begin{pmatrix} 0 & 1 & \infty & -2 & \infty \\ \infty & 0 & -1 & 3 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 4 \\ 3 & 4 & \infty & 1 & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 1 & 0 & -2 & \infty \\ \infty & 0 & -1 & 3 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 4 \\ 3 & 4 & 3 & 1 & 0 \end{pmatrix} \quad D_3 = \begin{pmatrix} 0 & 1 & 0 & -2 & \infty \\ \infty & 0 & -1 & 1 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 4 \\ 3 & 4 & 3 & 1 & 0 \end{pmatrix}$$

$$D_4 = \begin{pmatrix} 0 & 1 & 0 & -2 & 2 \\ \infty & 0 & -1 & 1 & 5 \\ \infty & \infty & 0 & 2 & 6 \\ \infty & \infty & \infty & 0 & 4 \\ 3 & 4 & 3 & 1 & 0 \end{pmatrix} \quad D_5 = \begin{pmatrix} 0 & 1 & 0 & -2 & 2 \\ 8 & 0 & -1 & 1 & 5 \\ 9 & 10 & 0 & 2 & 6 \\ 7 & 8 & 7 & 0 & 4 \\ 3 & 4 & 3 & 1 & 0 \end{pmatrix}$$



Υπολογισμός Συντομότερων Μονοπατιών

- $P_k[v_i, v_j]$ = προηγούμενη κορυφή της v_j στο συντομότερο $v_i - v_j$ μονοπάτι με ενδιάμεσες κορυφές μόνο από $V_k = \{v_1, \ldots, v_k\}$.
- $P_k[v_i, \cdot] = \Delta\Sigma\mathrm{M}(v_i)$ με ενδιάμεσες κορυφές από V_k.
- Αρχικά $P_0 \approx$ πίνακα γειτνίασης.

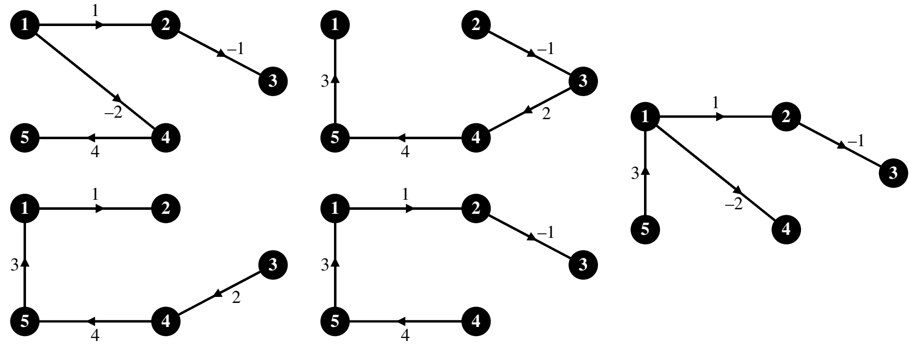
$$P_0[v_i, v_j] = \begin{cases} \mathrm{NIL} & \text{αν } i = j \text{ ή } (v_i, v_j) \notin E \\ v_i & \text{διαφορετικά} \end{cases}$$

- Αναδρομική σχέση για $P_0, P_1, \ldots, P_n$ με **δυναμικό προγραμματισμό**:

$$P_k[v_i, v_j] = \begin{cases} P_{k-1}[v_i, v_j] & \text{αν } D_{k-1}[v_i, v_j] \le D_{k-1}[v_i, v_k] + D_{k-1}[v_k, v_j] \\ P_{k-1}[v_k, v_j] & \text{αν } D_{k-1}[v_i, v_j] > D_{k-1}[v_i, v_k] + D_{k-1}[v_k, v_j] \end{cases}$$

- **Εύκολη τροποποίηση** αλγόριθμου Floyd-Warshall $\to$ **πίνακας** P.

Υπολογισμός Συντομότερων Μονοπατιών: Παράδειγμα



$$P_5 = \begin{pmatrix} \mathrm{NIL} & 1 & 2 & 1 & 4 \\ 5 & \mathrm{NIL} & 2 & 3 & 4 \\ 5 & 1 & \mathrm{NIL} & 3 & 4 \\ 5 & 1 & 2 & \mathrm{NIL} & 4 \\ 5 & 1 & 2 & 1 & \mathrm{NIL} \end{pmatrix}$$

Συμπεράσματα

- Συντομότερα μονοπάτια από **αρχική κορυφή** s:
  Συντομότερο $s - t$ μονοπάτι:
  - **Αρνητικά βάρη** $\rightarrow$ Bellman-Ford: $\Theta(nm)$.
    Δυναμικός Προγραμματισμός.
  - **Μη-αρνητικά βάρη** $\rightarrow$ Dijkstra: $\Theta(m + n \log n)$.
    Μέθοδος Απληστίας.
- Συντομότερα μονοπάτια για **όλα τα ζεύγη κορυφών**:
  - **Αρνητικά βάρη** $\rightarrow$ Floyd-Warshall: $\Theta(n^3)$.
    Δυναμικός Προγραμματισμός.
  - **Μη-αρνητικά βάρη** και $m = o(n^2)$ $\rightarrow$ Dijkstra: $\Theta(mn + n^2 \log n)$.
    Μέθοδος Απληστίας.