Συντομότερες Διαδρομές

Διδάσκοντες: **Σ. Ζάχος, Δ. Φωτάκης**

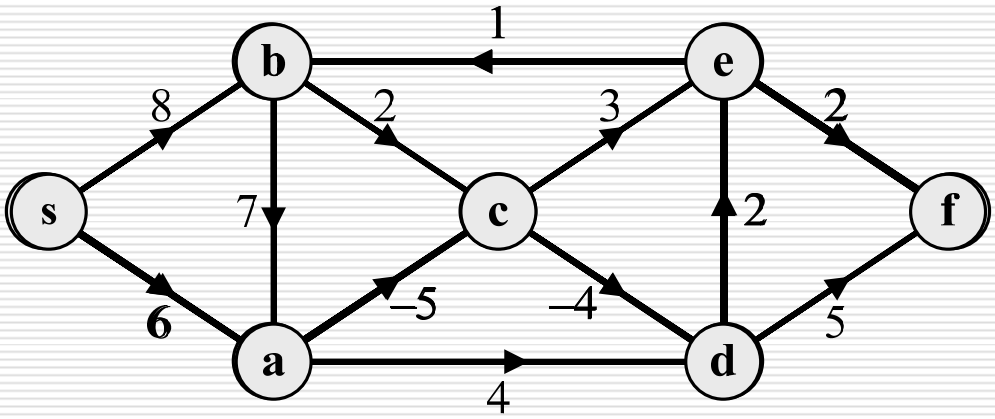
Επιμέλεια διαφανειών: **Δ. Φωτάκης**

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο

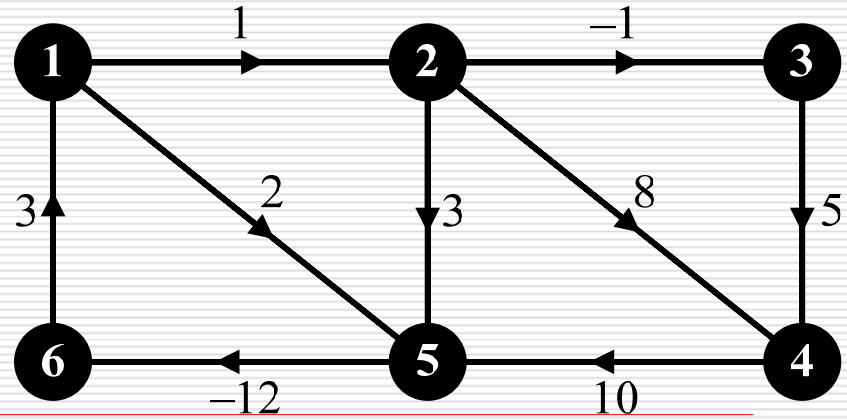
Συντομότερη Διαδρομή

- Κατευθυνόμενο G(V, E, w) με μήκη $w : E \mapsto \mathbb{R}$
  - Μήκος διαδρομής $p = (v_0, v_1, \ldots, v_k) : w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$
  - Απόσταση d(u, v): μήκος συντομότερης u – v διαδρομής.
  - Αν δεν υπάρχει u – v διαδρομή, d(u, v) = ∞.
- Ζητούμενο: αποστάσεις και συντομότερες διαδρομές από **αρχική κορυφή** s προς όλες τις κορυφές.
  - Θεμελιώδες πρόβλημα συνδυαστικής βελτιστοποίησης.



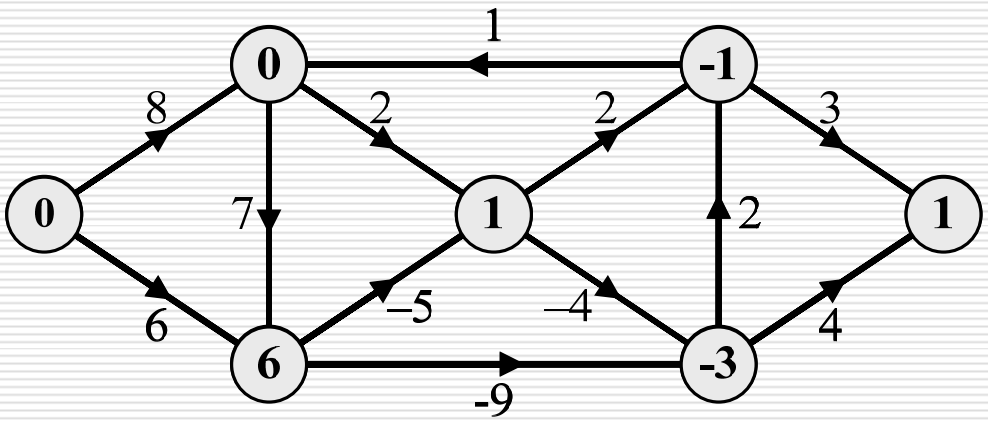
Κύκλοι Αρνητικού Μήκους

- Διαδρομή: ακολ. κορυφών όπου διαδοχικές συνδέονται με ακμή.
- Μονοκονδυλιά: διαδρομή χωρίς επαναλαμβανόμενες ακμές.
- (Απλό) μονοπάτι: διαδρομή χωρίς επαναλαμβανόμενες κορυφές.
  - Υπάρχει διαδρομή u – v ανν υπάρχει μονοπάτι u – v.
- Συντομότερη διαδρομή είναι **μονοπάτι** εκτός αν...
  - Υπάρχει κύκλος αρνητικού μήκους!
  - Αποστάσεις δεν ορίζονται γιατί συνολικό μήκος διαδρομής μπορεί να μειώνεται επ' άπειρο!
  - Κύκλος αρνητικού μήκους σε κάποια u – v διαδρομή $\Rightarrow d(u, v) = -\infty$.



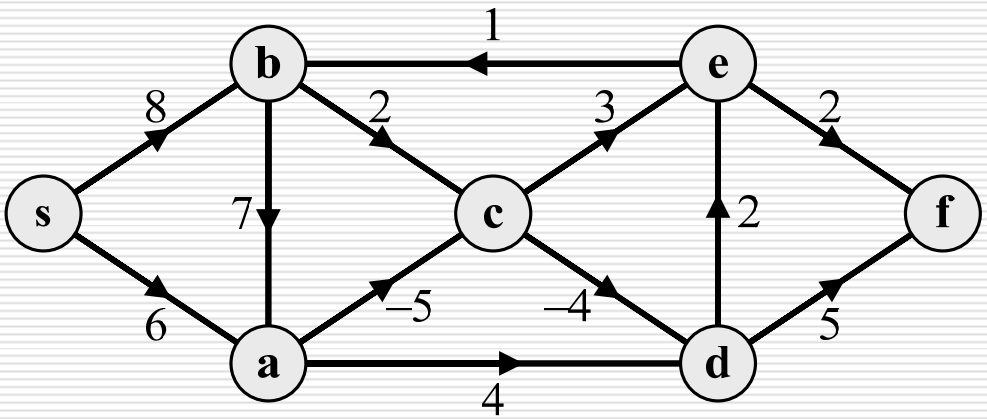
Συντομότερα Μονοπάτια

- Αν $p = (v_0, v_1, \ldots, v_k)$ είναι συντομότερο μονοπάτι, κάθε $v_i - v_j$ τμήμα του αποτελεί συντομότερο $v_i - v_j$ μονοπάτι.
  - Αρχή **βελτιστότητας**.
- Συντομότερα μονοπάτια από s προς όλες τις κορυφές: Δέντρο Συντομότερων Μονοπατιών (SPT, ΔΣΜ).
  - Αν συντομότερα $s - v_1$ και $s - v_2$ μονοπάτια έχουν κοινή κορυφή u, χρησιμοποιούν (ίδιο) συντομότερο s – u μονοπάτι.
  - ΔΣΜ αναπαρίσταται με **πίνακα προγόνων.**



Συντομότερα Μονοπάτια

- Ταυτίζεται ΔΣΜ με ΕΣΔ;
- Έστω συντομότερα μονοπάτια από s σε G(V, E, w).
  - Τι συμβαίνει σε G(V, E, kw), k > 0;
  - Τι συμβαίνει σε G(V, E, kw), k < 0;
  - Τι συμβαίνει σε G(V, E, w+k), k > 0;

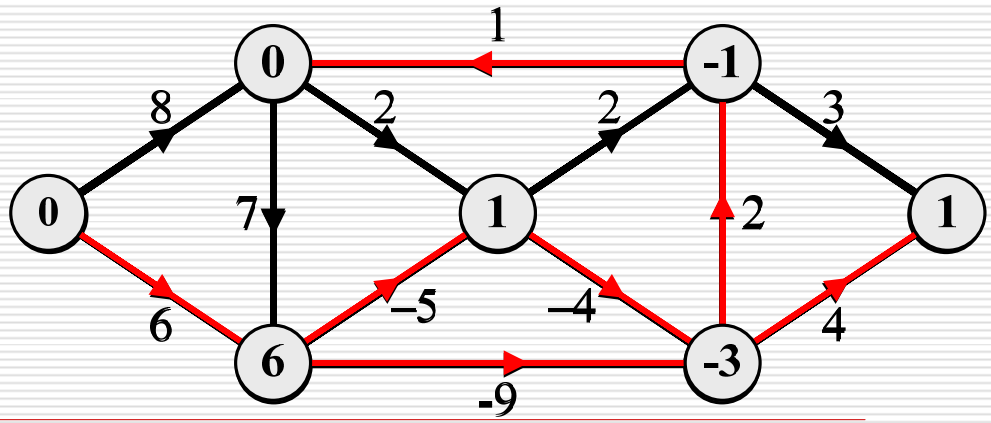


Αποστάσεις

- Αποστάσεις ικανοποιούν την «τριγωνική ανισότητα»:

$$\forall (v,u) \in E,\ d(s,u) \le d(s,v) + w(v,u)$$
$$\forall v,u \in V,\quad d(s,u) \le d(s,v) + d(v,u)$$

  - Ισότητα ισχύει ανν συντ. s – u μονοπάτι περιέχει ακμή (v, u) (αντίστοιχα, διέρχεται από κορυφή v).



Υπολογισμός Συντομότερων Μονοπατιών

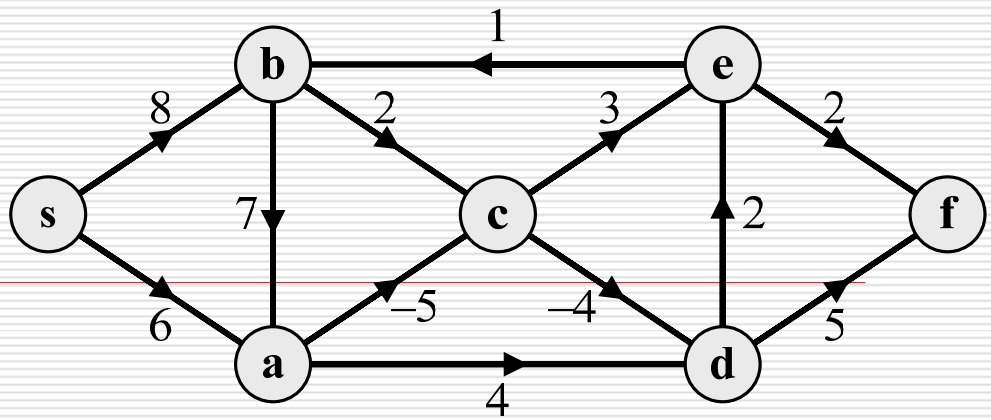
- Διατηρούμε «απαισιόδοξη» εκτίμηση D[u] για d(s, u).
  - Αρχικά: $D[s] = 0$ και $D[u] = \infty \quad \forall u \in V \setminus \{s\}$
    $p[u] = \text{NULL} \quad \forall u \in V$
  - Αλγόριθμος εξετάζει ακμές (v, u) και αναπροσαρμόζει D[u].
    Αν $D[u] > D[v] + w(v, u)$, τότε $D[u] \leftarrow D[v] + w(v, u)$
    $p[u] \leftarrow v$
- D[u] = μήκος συντομότερου γνωστού s – u μονοπατιού.
  - Επαγωγικά: αν ισχύει πριν τελευταία εξέταση ακμής (v, u), ισχύει και μετά αφού $D[u] \leftarrow \min\{D[u], D[v] + w(v, u)\}$
  - Πάντα $D[u] \geq d(s, u)$, και $D[u] = \infty$ αν $\nexists\, s - u$ μονοπάτι.
  - Όταν ακμές συντομότερου s – v μονοπατ. εξεταστούν με τη σειρά, γίνεται D[u] = d(s, u) και δεν μειώνεται στο μέλλον.
- Συστηματική εξέταση ακμών και κριτήριο τερματισμού.

Αλγόριθμος Bellman-Ford

- Ιδέα: δοκιμή όλων των ακμών σε κάθε πιθανή θέση για συντομότερο s – u μονοπάτι (ταυτόχρονα για όλες τις u).
  - D(u, i) = μήκος συντομότερου s – u μονοπ. με ≤ i ακμές.
  - Αρχικά D(s, 0) = 0 και D(u, 0) = ∞ για κάθε u ≠ s.
  - Από ΣΜ με ≤ i ακμές σε ΣΜ με ≤ i+1 ακμές:

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

  - (Απλό) μονοπάτι έχει ≤ n – 1 ακμές ⇒ D(u, n–1) = d(s, u)
    D(u, n) < D(u, n–1) αν κύκλος αρνητικού μήκους.
  - Υπολογισμός τιμών D(u, i), u ∈ V, i = 1, …, n, με **δυναμικό προγραμματισμό**.

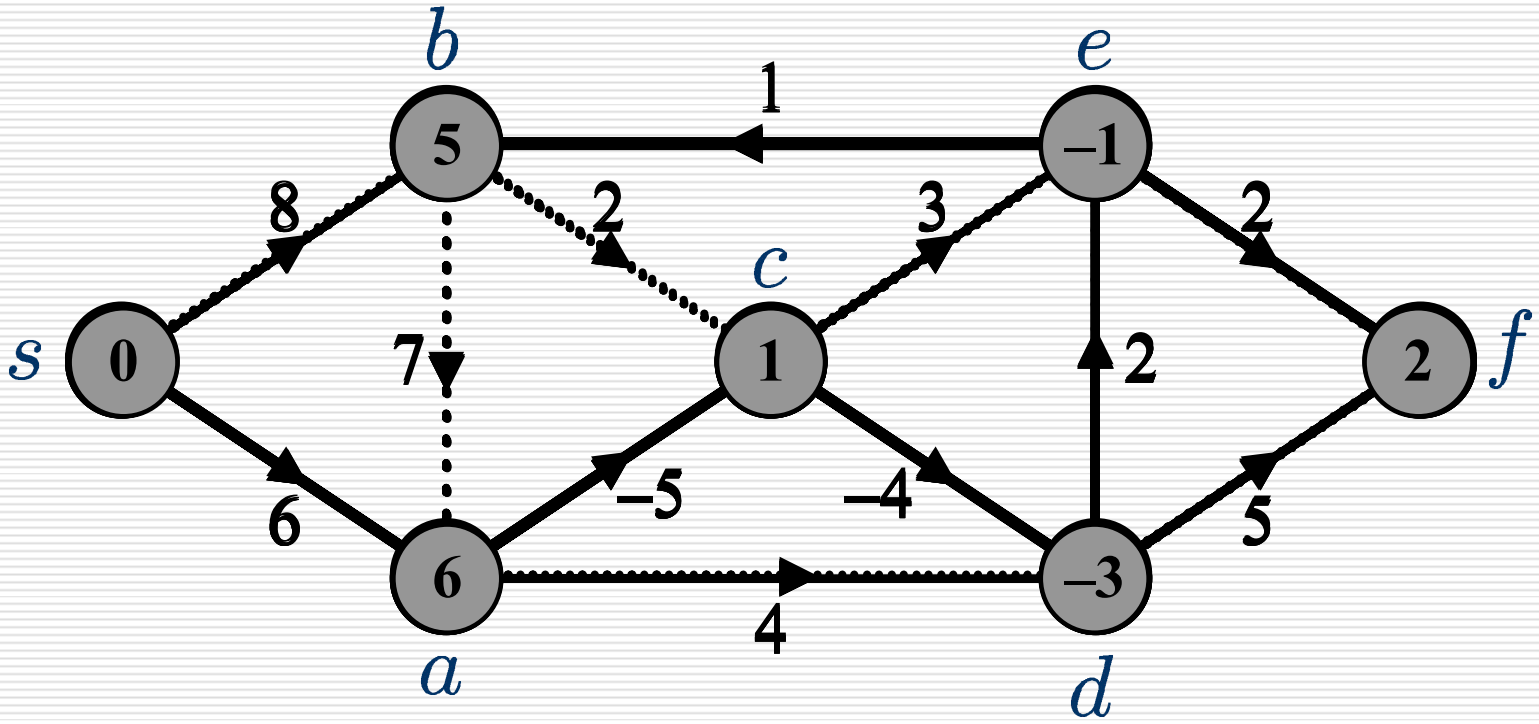


Αλγ. Bellman-Ford: Υλοποίηση

- «Απαισιόδοξη» εκτίμηση D[u].
  - Τέλος κάθε φάσης i, D[u] ≤ D[u, i]
- Σε φάση i = 1, …, n−1, κάθε ακμή εξετάζεται μία φορά.
- Επιπλέον φάση για έλεγχο ύπαρξης κύκλου αρνητικού μήκος.
- Χρόνος εκτέλεσης $\Theta(nm)$.

$\textbf{Bellman-Ford}(G(V,E,w),s)$
  $\textbf{for all } u \in V \textbf{ do}$
    $D[u] \leftarrow \infty;\ p[u] \leftarrow \text{NULL};$
  $D[s] \leftarrow 0;$
  $\textbf{for } i \leftarrow 1 \textbf{ to } n-1 \textbf{ do}$
    $\textbf{for all } (v,u) \in E \textbf{ do}$
      $\textbf{if } D[u] > D[v] + w(v,u) \textbf{ then}$
        $D[u] \leftarrow D[v] + w(v,u);$
        $p[u] \leftarrow v;$
  $\textbf{for all } (v,u) \in E \textbf{ do}$
    $\textbf{if } D[u] > D[v] + w(v,u) \textbf{ then}$
      $\textbf{return}(\text{NEG-CYCLE});$

Αλγ. Bellman-Ford: Παράδειγμα

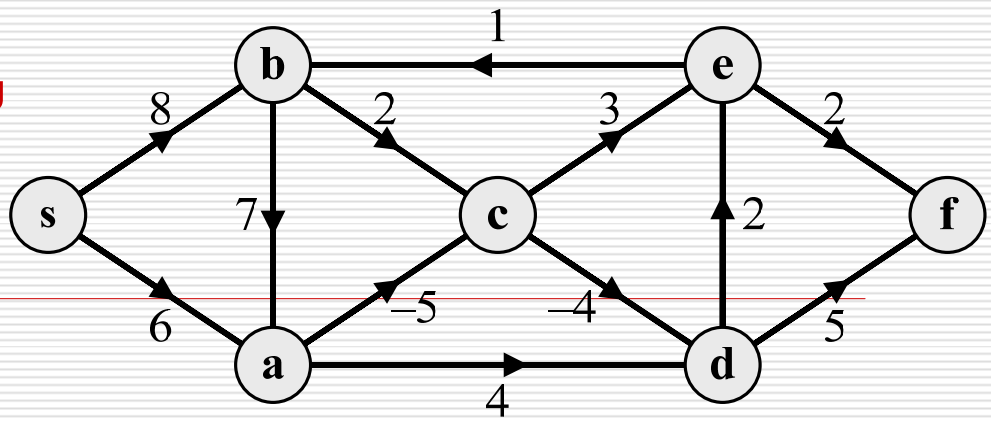


Αλγ. Bellman-Ford: Ορθότητα

- Αν όχι κύκλος αρνητικού μήκους, D[u] = d(s, u) στο τέλος.
  - Συντομότερο s – u μονοπάτι $s = v_0, v_1, \ldots, v_k = u$ με k ακμές.
  - Επαγωγική υπόθ.: Τέλος φάσης i – 1, $D[v_{i-1}] = d(s, v_{i-1})$.
  - Τέλος φάσης i: εξέταση ακμής (v_{i-1}, v_i) και $D[v_i] = d(s, v_i)$:

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

  - Τέλος φάσης n – 1: D[u] = d(s, u) για κάθε κορυφή u.
  - D[u] δεν μειώνεται άλλο, αφού πάντα D[u] ≥ d(s, u).
  - Αλγόριθμος δεν επιστρέφει ένδειξη για κύκλο αρνητικού μήκους.

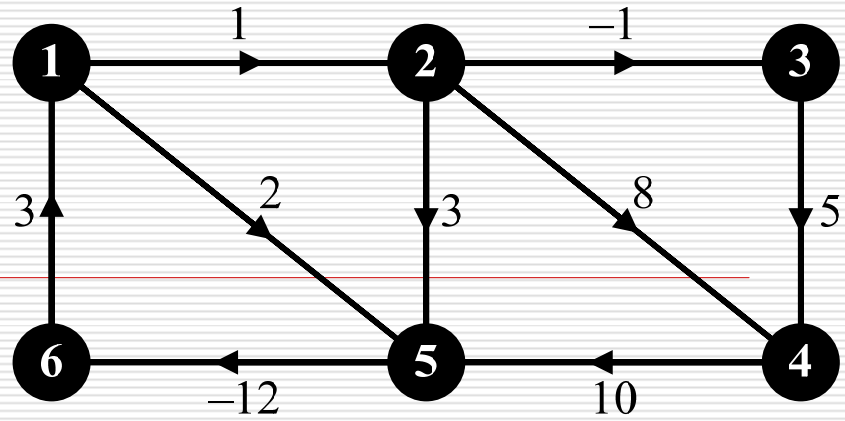


Αλγ. Bellman-Ford: Ορθότητα

- Αν κύκλος αρνητικού μήκους, ένδειξη στο τέλος.
  - Έστω κύκλος αρνητικού μήκους $v_0, v_1, \ldots, v_{k-1}, v_k$ (= v_0) προσπελάσιμος από s.
  - Εκτιμήσεις $D[v_i]$ πεπερασμένες στο τέλος φάσης n−1.
  - Αν όχι ένδειξη, πρέπει στη φάση n για κάθε v_i στον κύκλο: $D[v_i] \leq D[v_{i-1}] + w(v_{i-1}, v_i)$
  - Αθροίζοντας κατά μέλη:

$$\sum_{i=1}^{k} D[v_i] \leq \sum_{i=1}^{k} D[v_{i-1}] + \sum_{i=1}^{k} w(v_{i-1}, v_i) \Rightarrow \sum_{i=1}^{k} w(v_{i-1}, v_i) \geq 0$$

  - Άτοπο! Άρα ο αλγόριθμος επιστρέφει ένδειξη για κύκλο αρνητικού μήκους.

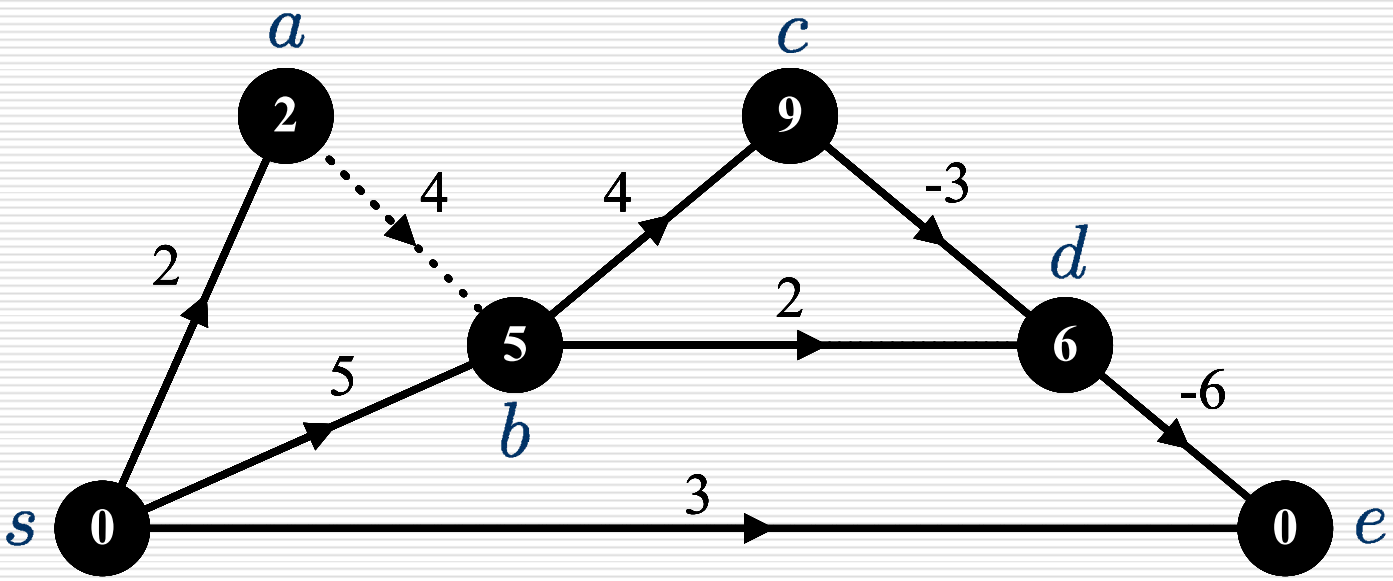


Συντομότερα Μονοπάτια σε DAG

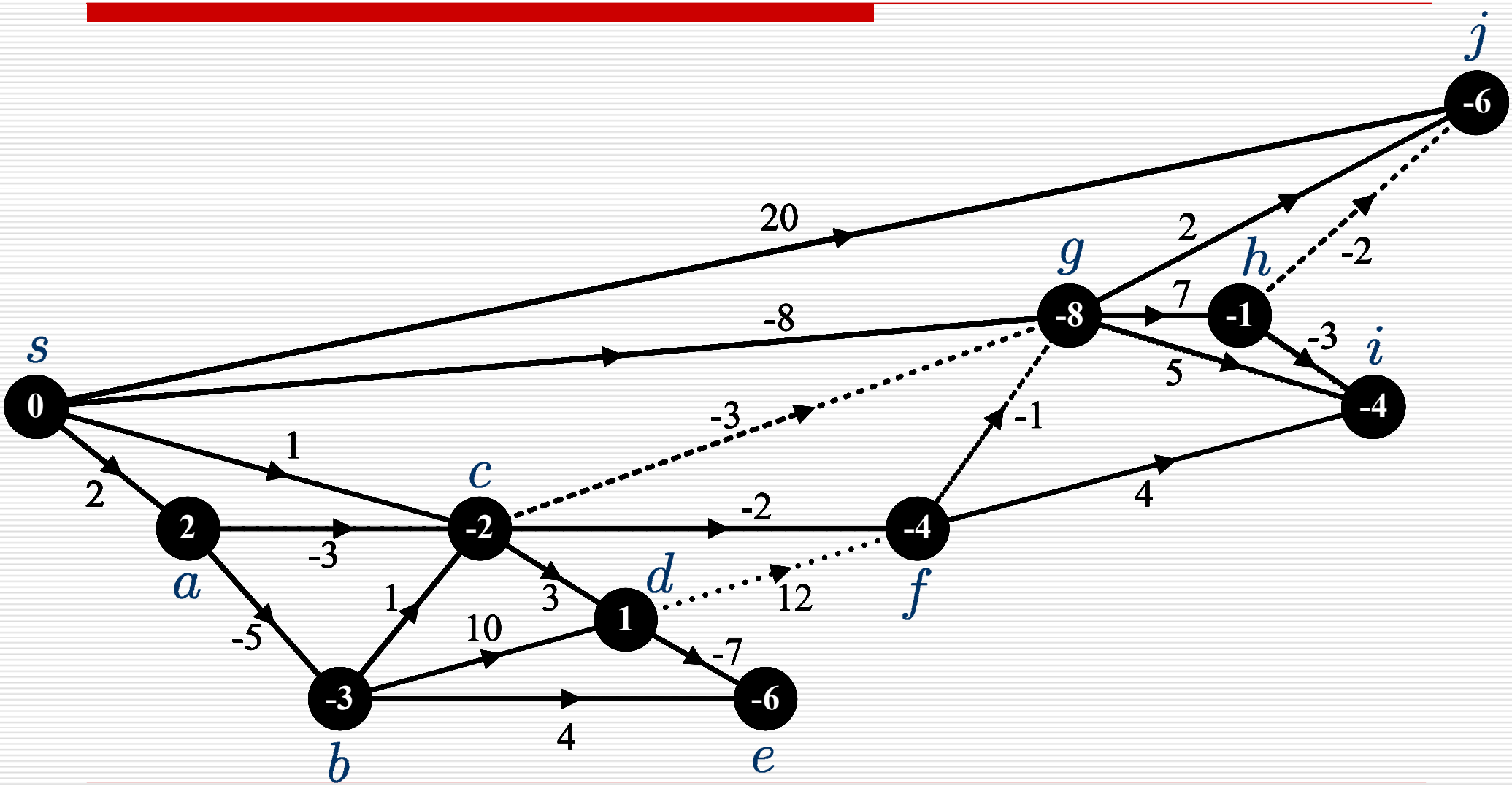
- Σε DAG, σειρά εμφάνισης κορυφών σε κάθε μονοπάτι (άρα και ΔΣΜ) ακολουθεί τοπολογική διάταξη!
  - Έστω τοπολογική διάταξη $s = v_1, v_2, \ldots, v_n$.
  - $d(s, v_k)$ εξαρτάται μόνο από $d(s, v_j)$ με $j < k$: $$d(s,v_k) = \min_{v_j:(v_j,v_k)\in E} \{d(s,v_j) + w(v_j,v_k)\}$$
- Κορυφές εντάσσονται στο ΔΣΜ με σειρά τοπολογ. διάταξης και εξετάζονται εξερχόμενες ακμές τους (μια φορά κάθε ακμή!).
  - Ορθότητα με επαγωγή (παρόμοια με Bellman-Ford).
  - Επαγωγική υπόθ.: ακριβώς πριν την ένταξη του v_k στο ΔΣΜ, ισχύει ότι $D[v_j]=d(s,v_j)$ για κάθε $j = 0, \ldots, k$.
  - Ακριβώς πριν ένταξη v_{k+1} στο ΔΣΜ, $D[v_{k+1}]=d(s,v_{k+1})$ αφού

$$D[v_{k+1}] = \min_{v_j:(v_j,v_{k+1})\in E} \{D[v_j] + w(v_j,v_{k+1})\}$$

Παράδειγμα



Παράδειγμα



Συντομότερα Μονοπάτια σε DAG

- Χρόνος εκτέλεσης: γραμμικός, Θ(n+m)
- Χρησιμοποιείται και για υπολογισμό μακρύτερων μονοπατιών.
  - Αν G(V, E, w) **ακυκλικό**, p μακρύτερο s – u μονοπάτι αν p συντομότερη s – u διαδρομή στο G(V, E, –w).

ShortestPath-DAG($G(V, E, w), v_1$)

```
Έστω τοπολογική διάταξη v_1, v_2, ..., v_n;
for j ← 1 to n do
    D[v_j] ← ∞; p[v_j] ← NULL;
D[v_1] ← 0;
for j ← 1 to n − 1 do
    for all (v_j, v_i) ∈ E do
        if D[v_i] > D[v_j] + w(v_j, v_i) then
            D[v_i] ← D[v_j] + w(v_j, v_i);
            p[v_i] ← v_j;
```

Αλγόριθμος Dijkstra

- Ταχύτερα αν όχι αρνητικά μήκη! Αποτελεί γενίκευση BFS.
  - Ταχύτερα αν υπάρχει πληροφορία για σειρά εμφάνισης κορυφών σε συντομότερα μονοπάτια (και ΔΣΜ).
  - Μη αρνητικά μήκη: κορυφές σε αύξουσα σειρά απόστασης.
- Κορυφές εντάσσονται σε ΔΣΜ σε αύξουσα απόσταση και εξετάζονται εξερχόμενες ακμές τους (μια φορά κάθε ακμή!).
  - Αρχικά D[s] = 0 και D[u] = ∞ για κάθε u ≠ s.
  - Κορυφή u εκτός ΔΣΜ με ελάχιστο D[u] εντάσσεται σε ΔΣΜ.
  - Για κάθε ακμή (u, v), $D[v] \leftarrow \min\{D[v], D[u] + w(u, v)\}$
- Ορθότητα: όταν u εντάσσεται σε ΔΣΜ, D[u] = d(s, u).
  - Μη αρνητικά μήκη: κορυφές v με μεγαλύτερο D[v] σε μεγαλύτερη απόσταση και δεν επηρεάζουν D[u].

Αλγόριθμος Dijkstra

- Άπληστος αλγόριθμος.
- **Υλοποίηση**:
  - Ελάχιστο D[v]: ουρά προτεραιότητας.
  - Binary heap: $\Theta(m \log n)$
  - Fibonacci heap: $\Theta(m + n \log n)$
  - Ελάχιστο D[v] γραμμικά: $\Theta(n^2)$.

$\text{Dijkstra}(G(V, E, w), s)$

  for all $u \in V$ **do**

    $D[u] \leftarrow \infty;\ \ p[u] \leftarrow \text{NULL};$

  $D[s] \leftarrow 0;\ \ S \leftarrow \emptyset;$

  while $|S| < |V|$ **do**

    $u \notin S : D[u] = \min_{v \notin S}\{D[v]\};$

    $S \leftarrow S \cup \{u\};$

    for all $v \in \text{AdjList}[u]$ **do**

      if $D[v] > D[u] + w(u, v)$ **then**

        $D[v] \leftarrow D[u] + w(u, v);$

        $p[v] \leftarrow u;$

Κάτι μου Θυμίζει ...;!

Dijkstra($G(V, E, w), s$)

for all $u \in V$ **do**

$D[u] \leftarrow \infty;\ \ p[u] \leftarrow$ NULL;

$D[s] \leftarrow 0;\ \ S \leftarrow \emptyset;$

while $|S| < |V|$ **do**

$u \notin S : D[u] = \min_{v \notin S}\{D[v]\};$

$S \leftarrow S \cup \{u\};$

for all $v \in \text{AdjList}[u]$ **do**

if $D[v] > D[u] + w(u, v)$ **then**

$D[v] \leftarrow D[u] + w(u, v);$

$p[v] \leftarrow u;$

MST-Prim($G(V, E, w), s$)

for all $u \in V$ **do**

$c[u] \leftarrow \infty;\ \ p[u] \leftarrow$ NULL;

$c[s] \leftarrow 0;\ \ S \leftarrow \emptyset;\ \ \Delta \leftarrow \emptyset;$

while $|S| < |V|$ **do**

$u \notin S : c[u] = \min_{v \notin S}\{c[v]\};$

$S \leftarrow S \cup \{u\};$

for all $v \in \text{AdjList}[u]$ **do**

if $v \notin S$ **and** $w(u, v) < c[v]$ **then**

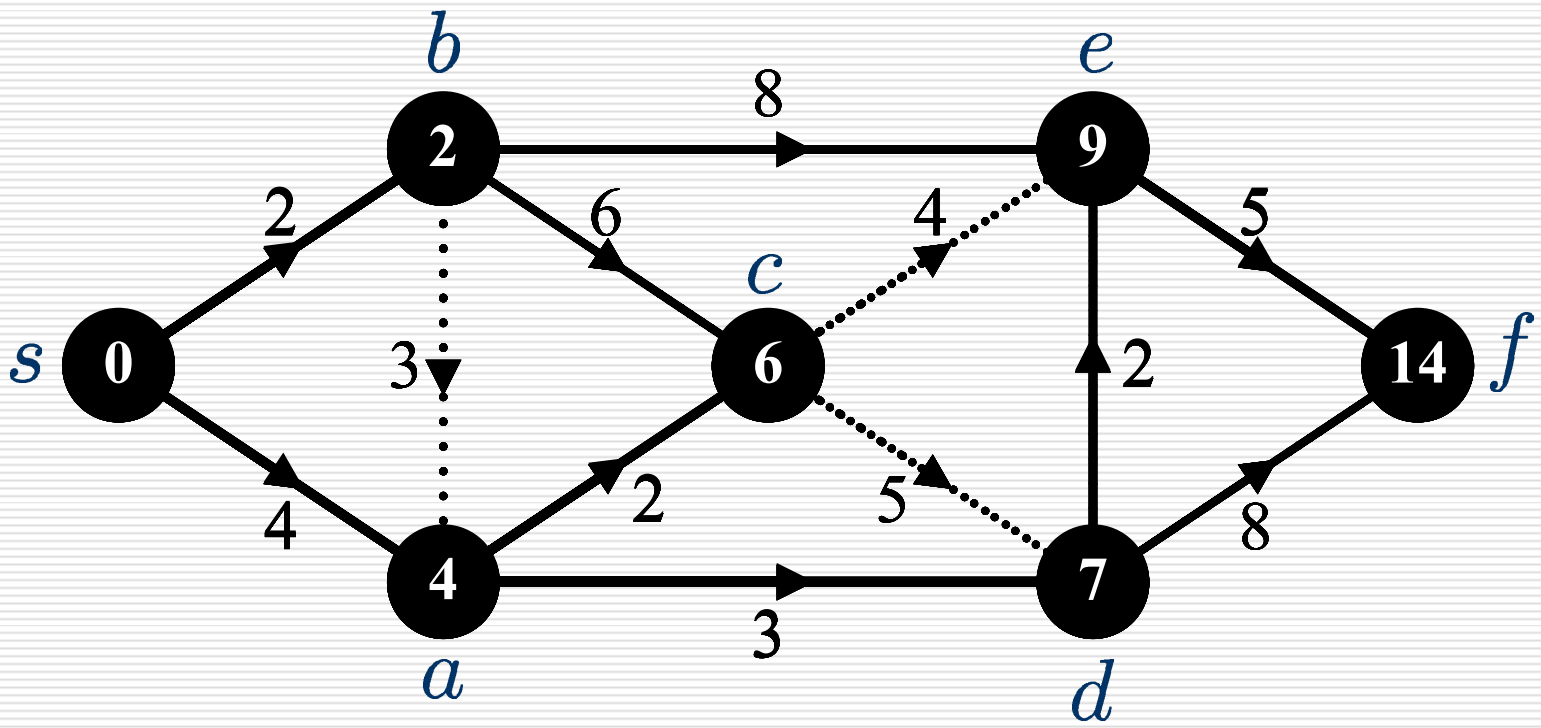
$c[v] \leftarrow w(u, v);$

$p[v] \leftarrow u;$

if $p[u] \neq$ NULL **then**

$\Delta \leftarrow \Delta \cup \{u, p[u]\};$

Αλγόριθμος Dijkstra: Παράδειγμα



Αλγόριθμος Dijkstra: Εξέλιξη

- Απόσταση (και συντομότερο μονοπάτι) από s προς κοντινότερη (στην s), 2η κοντινότερη (στην s) κορυφή, κοκ.
- Συντομότερα μονοπάτια για κοντινότερες κορυφές, με υπολογισμένες αποστάσεις, σχηματίζουν υποδέντρο του ΔΣΜ.
- Επόμενη κοντινότερη (στην s) κορυφή είναι **συνοριακή** κορυφή.
  - Συνοριακή κορυφή: δεν ανήκει σε υποδέντρο ΔΣΜ και έχει εισερχόμενη ακμή από υποδέντρο.
- Εκτιμήσεις απόστασης συνοριακών κορυφών διατηρούνται σε ουρά προτεραιότητας.
- Συνοριακή κορυφή με ελάχιστη εκτίμηση απόστασης «βγαίνει» από ουρά προτεραιότητας και προστίθεται στο υποδέντρο.
  - Εκτιμήσεις απόστασης συνοριακών κορυφών ενημερώνονται με προσθήκη νέας κορυφής στο υποδέντρο (για εξερχόμενες ακμές της).

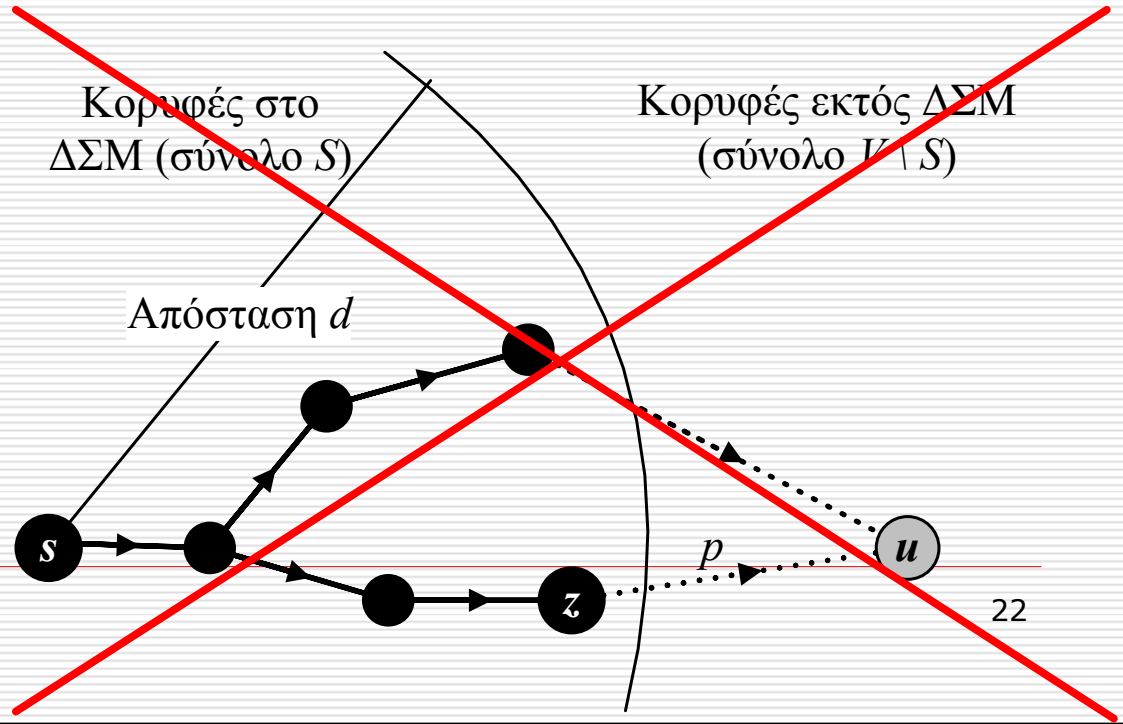
Αλγόριθμος Dijkstra: Ορθότητα

- Θ.δ.ο όταν κορυφή u εντάσσεται σε ΔΣΜ, D[u] = d(s, u).
  - Επαγωγή: έστω D[v] = d(s, v) για κάθε v ήδη στο ΔΣΜ.
  - u έχει ελάχιστο D[u] (εκτός ΔΣΜ). Έστω ότι D[u] > d(s, u).
  - p συντομότερο s – u μονοπάτι με μήκος d(s, u) < D[u], και z τελευταία κορυφή πριν u στο p:

Μπορεί z στο ΔΣΜ;

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

Όχι!



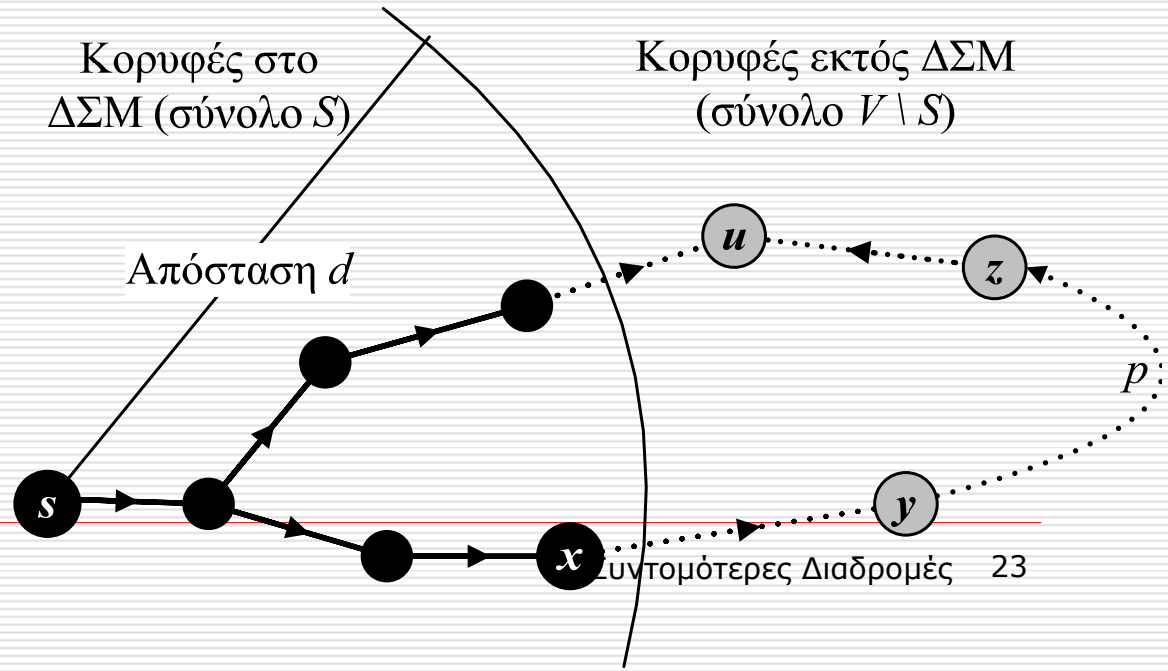
Αλγόριθμος Dijkstra: Ορθότητα

- Θ.δ.ο όταν κορυφή u εντάσσεται σε ΔΣΜ, D[u] = d(s, u).
  - Επαγωγή: έστω D[v] = d(s, v) για κάθε v ήδη στο ΔΣΜ.
  - u έχει ελάχιστο D[u] (εκτός ΔΣΜ). Έστω ότι D[u] > d(s, u).
  - p συντομότερο s – u μονοπάτι με μήκος d(s, u) < D[u], και z τελευταία κορυφή πριν u στο p:

Έστω x (≠ z) τελευταία κορυφή του p στο ΔΣΜ και y (μπορεί y = z) επόμενη της x στο p.

$$D[y] \le D[x] + w(x,y)$$
$$= d(s,x) + w(x,y)$$
$$= d(s,y) < D[u]$$

$\Rightarrow D[y] < D[u]$, **άτοπο!**



Dijkstra vs Bellman-Ford

- Αλγ. Dijkstra ταχύτερος κατά n αλλά δεν εφαρμόζεται για αρνητικά μήκη.
  - Βασίζεται στο ότι αποστάσεις δεν μειώνονται κατά μήκος συντομότερου μονοπατιού.
- Αλγ. Bellman-Ford εφαρμόζεται για αρνητικά μήκη.
  - Αποστάσεις μπορεί να μειώνονται κατά μήκος συντομότερου μονοπατιού.
  - «Τελευταία» κορυφή μπορεί σε μικρότερη απόσταση από αρχική.

Ερωτήσεις – Ασκήσεις

- Αρνητικά μήκη $\rightarrow$ προσθέτουμε μεγάλο αριθμό $\rightarrow$
  $\rightarrow$ θετικά μήκη $\rightarrow$ αλγόριθμος Dijkstra;
- Ndo BFS υπολογίζει ΔΣΜ όταν ακμές μοναδιαίου μήκους.
- Όταν μη-αρνητικά μήκη, μπορεί ένα ΔΣΜ και ένα ΕΣΔ να μην έχουν καμία κοινή ακμή;
- **Bottleneck** Shortest Paths:
  - Κόστος μονοπατιού p: $c(p) = \max_{e \in p}\{w(e)\}$
  - Υπολογισμός ΔΣΜ για bottleneck κόστος;
  - Τροποποίηση Dijkstra λύνει Bottleneck Shortest Paths (ακόμη και για αρνητικά μήκη):

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