Συντομότερα Μονοπάτια

- Απόσταση ή κόστος ή βάρος μονοπατιού σε βεβαρημένα γραφήματα:
  - Άθροισμα των βαρών των εμπλεκομένων ακμών
- Ανάγκη υπολογισμού των συντομότερων *απλών* μονοπατιών
- Τριγωνική ανισότητα:

$$w(x,y) \leq w(x,z) + w(z,y),\ (x,y),(x,z),(z,y) \in E$$

- Τεχνική ***χαλαρώσεως ακμών*:**
  - Δίδονται προσεγγίσεις για τις αποστάσεις, οι οποίες, προϊόντος του χρόνου, βελτιώνονται, δοκιμάζοντας μήπως η χρήση κάποιας ακμής «χαλαρώνει» την αρχική εκτίμηση για την κορυφή-απόληξη
- Τεχνική ***χαλαρώσεως μονοπατιού*:**
  - η χρήση κάποιας ακμής «χαλαρώνει» την αρχική εκτίμηση για ένα μονοπάτι

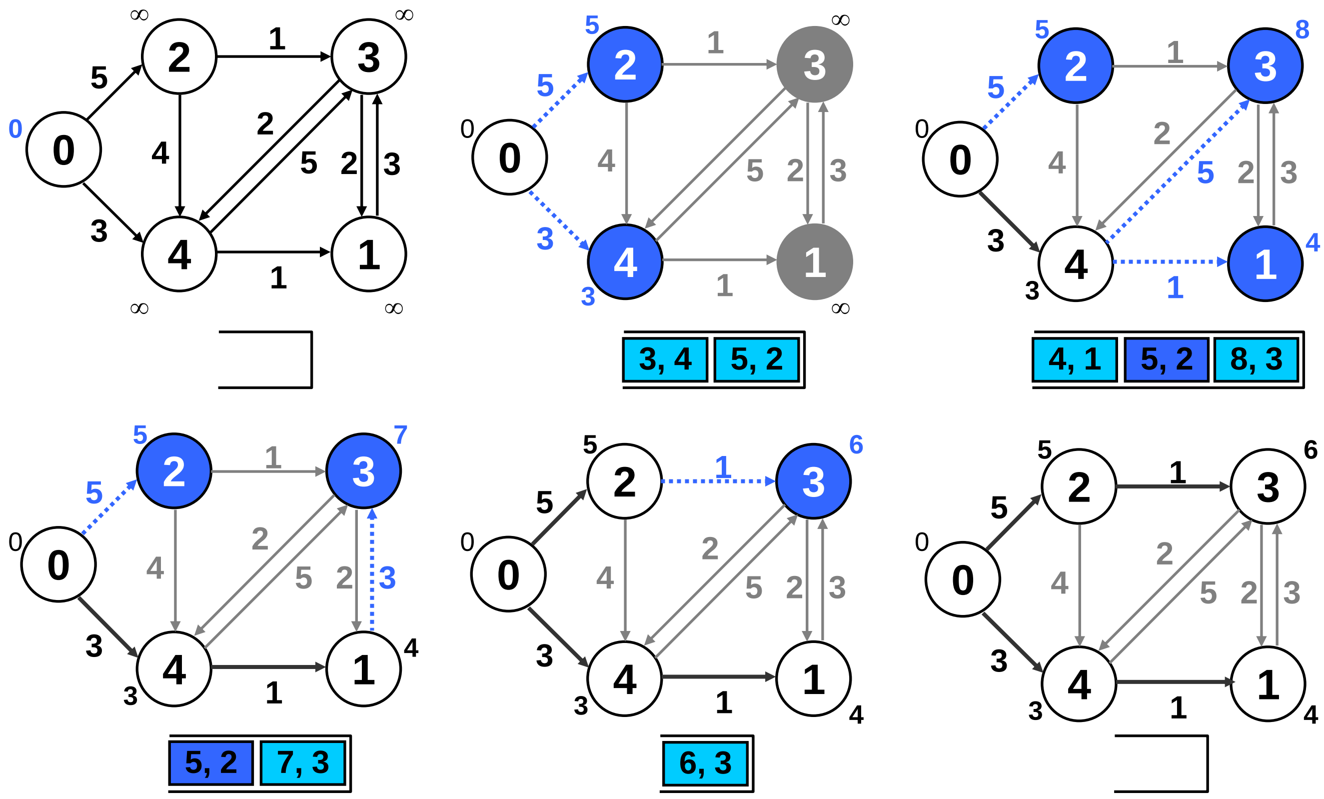
Συντομότερα Μονοπάτια Μοναδικής Πηγής (ΣΜΜΠ)

- **Το Ζητούμενο**
  - Εύρεση όλων των συντομότερων μονοπατιών από μία κορυφή-πηγή (source) *v* προς τις υπόλοιπες
  - Τα ανωτέρω μονοπάτια σχηματίζουν το ***δένδρο ΣΜΜΠ*** με ρίζα την κορυφή *v*

Αλγόριθμος του Dijkstra

- Κλασσικό παράδειγμα αλγορίθμου που βασίζεται στην *αρχή της απληστίας*
- Μόνο για βεβαρημένα γραφήματα ***θετικού βάρους (κόστους)***
- Διατηρεί μια διαμέριση του συνόλου *V* των κορυφών:
  - σε αυτές που ξέρουμε ακριβώς την απόσταση (*S*), και
  - σε αυτές που η γνώση μας είναι ελλιπής και προσεγγιστική (*V*-*S*)
- Επαναληπτικά,
  - αφαιρεί την «κοντινότερη» κορυφή *v* από το δεύτερο σύνολο,
  - εντάσσει την *v* στο πρώτο σύνολο,
  - διενεργεί χαλαρώσεις επί των εξερχόμενων ακμών της *v* με απολήξεις κορυφές του *V*-*S, ώστε το σύνολο παρυφής να είναι ενημερωμένο*

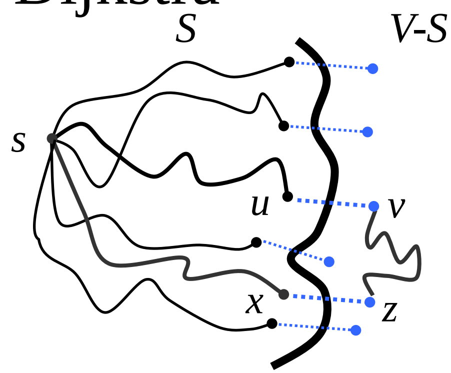
Παράδειγμα Dijkstra



Σημείωση. Για λόγους «οικονομίας» χώρου, στην ουρά δεν εικονίζονται οι αρχικές τιμές-προσέγγιση ∞ προς όλες τις κορυφές

Απόδειξη Ορθότητας

- Βασίζεται στην απουσία αρνητικών ακμών:
  – Όταν μία κορυφή *v* επιλέγεται ως η επόμενη που θα προστεθεί, τότε *πράγματι η απόσταση της από την πηγή έχει συγκλίνει στην τελική τιμή*
  – Έστω $s \in u \rightarrow v$ το μονοπάτι που επιστρέφει ο αλγόριθμος. Ας υποθέσουμε ότι υπήρχε άλλο συντομότερο μονοπάτι για την *v*: $s \in x \rightarrow z \in v$, το οποίο περιλαμβάνει και κορυφές που εξετάζει ο Dijkstra αργότερα.
  – Άτοπο, γιατί:
    - dis($s \in u \rightarrow v$) ≤ dis($s \in x \rightarrow z$) (λόγω τρεξίματος)
    - dis($z \in v$) > 0 (θετικά βάρη)



Ψευδοκώδικας

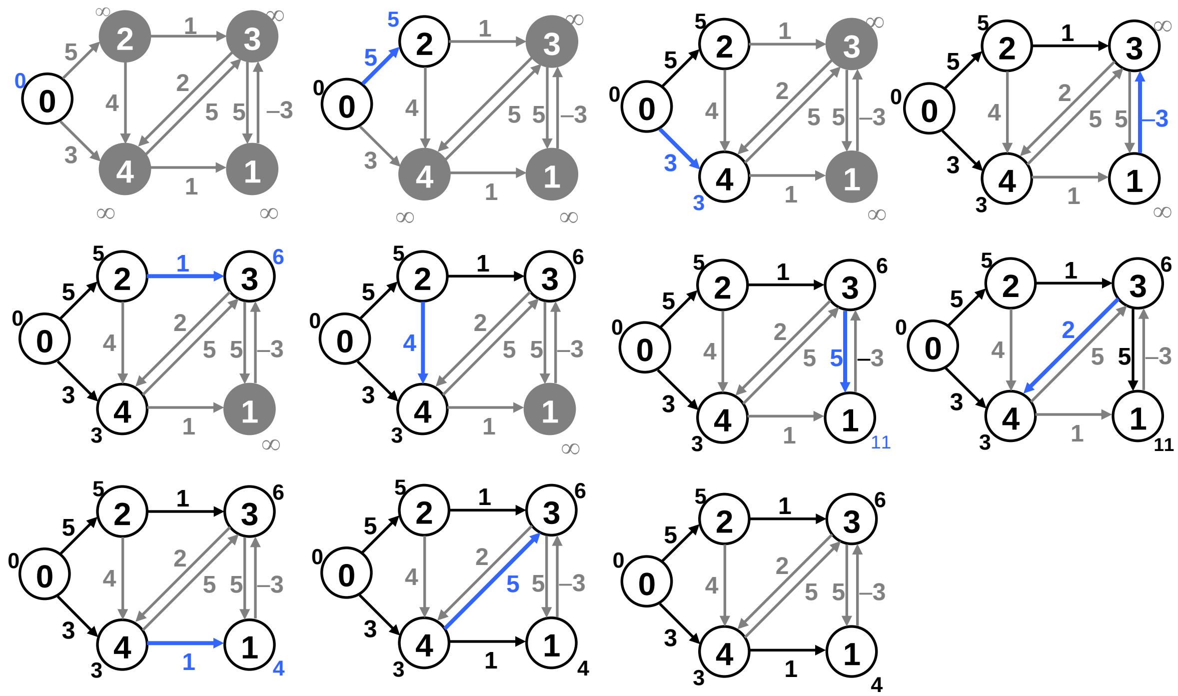
```
Algorithm DijkstraSSSP(graph g, double dis[], vertex s)
pqueue pq = new pqueue();            // η ουρά προτεραιότητας
for (v = 0; v < g.V; v++){           // αρχικοποίηση
   dis[v] = INFTY;
   pq.insKey(v,dis[v]);
}
dis[s] = 0.0;
pq.decKey(s,dis[s]);
while (!pq.isEmpty()) {              // όσο υπάρχουν κορυφές παρυφής
  v = pq.delmin();                   // εκτελείται #V φορές
  for (x = g.List[v]; x != null; x = x.getNext())  // έλεγχος των γειτονικών κορυφών της v
     if ((dis[v] + x.weight) < dis[w = x.v]){
       dis[w] = dis[v] + x.weight;   // χαλάρωση
       pq.decKey(w,dis[w]);          // ενημέρωση ουράς: #E φορές, το πολύ
       g.T[w] = v;                   // προσθήκη στο δένδρο
     }
}
```

- Ομοιότητα με Prim, όμως διαφορά στον στόχο (υπολογισμό αποστάσεων και όχι δένδρου)
- **Πολυπλοκότητα:** με δυωνυμική ουρά $O(V\log V + E\log V) = O(E\log V)$, με Fibonacci $O(V\log V + E)$

Αλγόριθμος Bellman-Ford

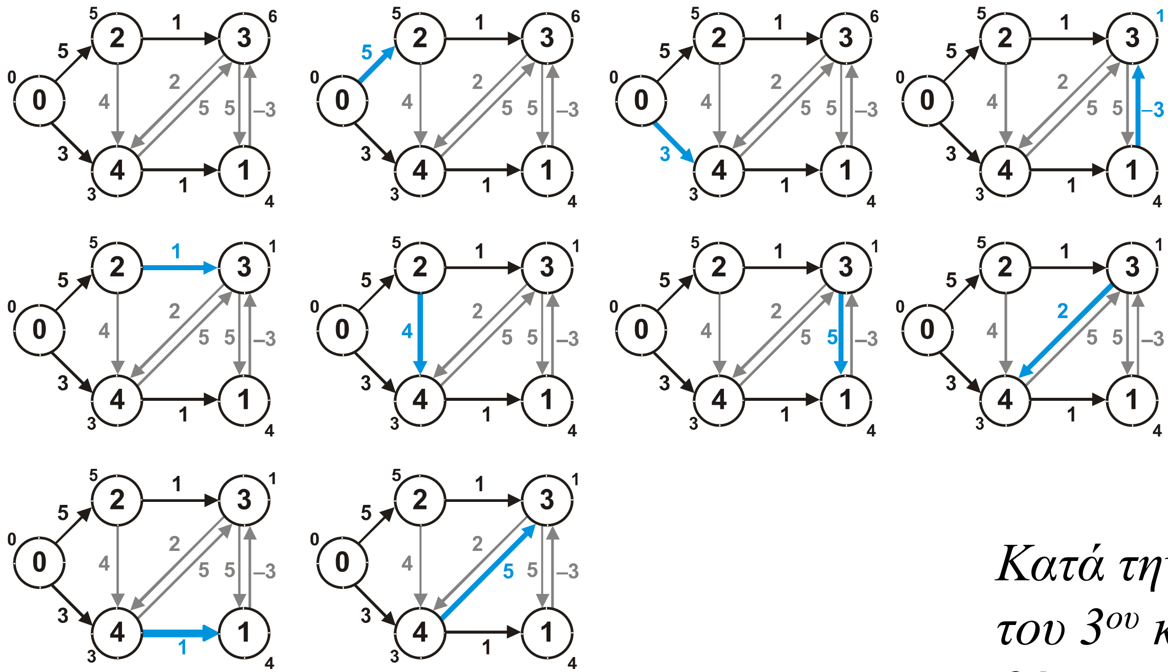
- Δουλεύει ***και*** με αρνητικά βάρη
- Στηρίζεται στην αρχή χαλαρώσεως:
  - Λόγω αρνητικών βαρών, είναι δυνατόν μία ακμή να χαλαρώσει περισσότερες από μία φορές
  - Γι’ αυτό και εκτελεί V-1 βήματα χαλάρωσης, εξετάζοντας ***όλες*** τις ακμές
  - Αν κατά το V-στό βήμα σημειωθεί χαλάρωση, τότε ***υπάρχει αρνητικός κύκλος***

Παράδειγμα Bellman-Ford

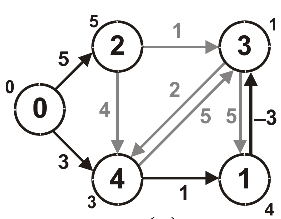


(α)

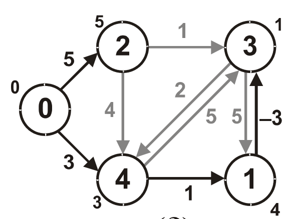
Παράδειγμα Bellman-Ford (συν.)



(β)



(γ)



(δ)

Κατά την διάρκεια του $3^{ου}$ και $4^{ου}$ βήματος χαλάρωσης (Σχ. (γ)-(δ)), δεν σημειώνεται καμμία αλλαγή

Απόδειξη Ορθότητας

- Με επαγωγή:
  – Κατά το *i*-στο βήμα, οι αποστάσεις που ανακαλύπτονται, δεν ξεπερνούν τα συντομότερα μονοπάτια το *πολύ i* ακμών.

    Απόδειξη. Έστω ότι το συντομότερο μονοπάτι από την *s* προς την *w*, το πολύ *i* ακμών, περνά από την (*v*,*w*). Κατά το (*i*+1)-στάδιο, δοκιμάζουμε χαλάρωση της (*v*,*w*)

    - *Καμμία αλλαγή*:
      – Τότε, βάσει επαγωγής, η dis(s,w) είναι το πολύ ίση με την συντομότερη απόσταση μονοπατιού το πολύ *i* ακμών, άρα και το πολύ *i*+1ακμών
    - *Επιτυχής χαλάρωση*:
      – Τότε, το συντομότερο μονοπάτι από την *s* στην *v* έχει το πολύ *i* ακμές - βάσει επαγωγής, η dis(*s*,*v*), αφού ολοκληρώθηκε κατά το *i*-στό στάδιο, πήρε τιμή που δεν ξεπερνά το συντομότερο μονοπάτι *i*, το πολύ, ακμών
      – Άρα, η απόσταση dis(*s*,*w*)= dis(*s*,*v*) + βάρος(*v*,*w*) δεν ξεπερνά την τιμή του συντομότερου μονοπατιού *i*+1, το πολύ, ακμών

Απόδειξη Ορθότητας (συν.)

– Επομένως, μετά τα *V*-1 στάδια, όλες οι αποστάσεις που έχουν υπολογιστεί δεν ξεπερνούν το μήκος των συντομότερων μονοπατιών *V*-1 ακμών, το πολύ (μέγιστο # ακμών)
– Εάν κατά το *V* στάδιο σημειωθεί κάποια μείωση, τότε σίγουρα υπάρχει αρνητικός κύκλος, αφού το μονοπάτι θα έχει > *V*-1 ακμές

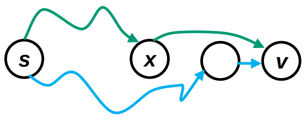
Ψευδοκώδικας

```
Algorithm BellmanFordSSSP(graph g, vertex s)
double[] dis = new double[g.V];
for (v = 0; v < g.V; v++){  // αρχικοποίηση
  dis[v] = INFTY;
  g.T[v] = -1;
}
dis[s] = 0;
for (i = 0; i < g.V-1; i++)          // Βήμα χαλάρωσης, #V-1 φορές συνολικά
  for (x = 0; x < g.E; x++)          // δοκιμή χαλάρωσης κάθε ακμής, #E φορές συνολικά
    if ((dis[v = g.Edges[x].v]+g.Edges[x].weight) < dis[w = g.Edges[x].w]){
      dis[w] = dis[v]+g.Edges[x].weight;
      g.T[w] = v;
    }
for (x = 0; x < g.E; x++)            // Δοκιμή για αρνητικό κύκλο
  if  ((dis[g.Edges[x].v]+g.Edges[x].weight) < dis[g.Edges[x].w])
    return "αρνητικός κύκλος";
return dis;
```

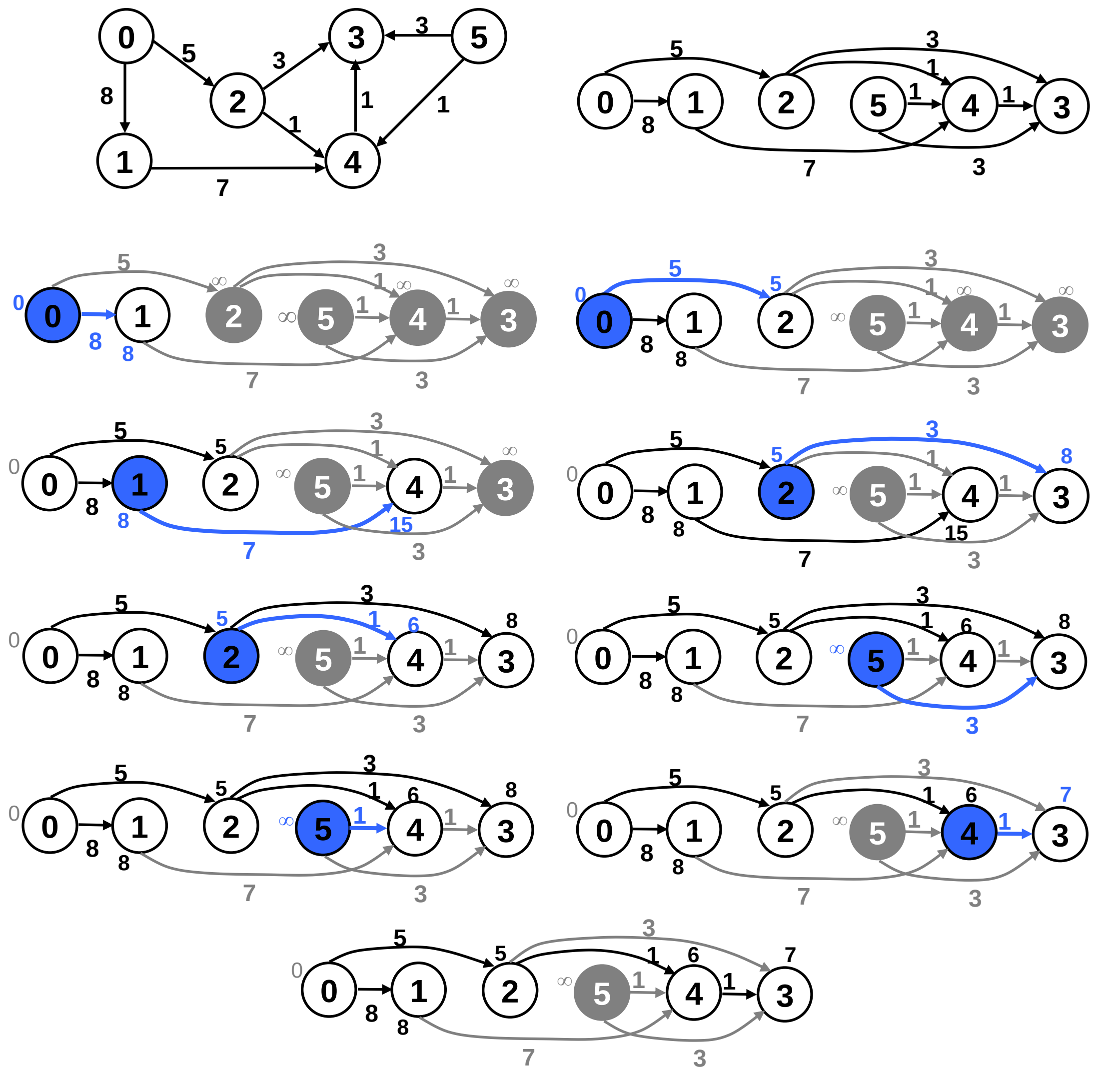
- **Πολυπλοκότητα: *O*(*VE*)**
- **Καλύτερη υλοποίηση με ουρά, γιατί μόνον οι ακμές που καταλήγουν σε κορυφές με ανανεωμένη πληροφορία έχουν πιθανότητα να βελτιώσουν την γνώση**

ΣΜΜΠ ΣΕ ΚΑΓ

- Τα ΚΑΓ επιτρέπουν, σε γραμμικό χρόνο, τον υπολογισμό ΣΜΜΠ με πολύ απλό τρόπο:
  – Οι κορυφές εξετάζονται κατά τοπολογική σειρά, δοκιμάζοντας για χαλάρωση όλες τις ακμές εξόδου τους
- Ορθότητα πηγάζει από την τοπολογική σειρά:
  – Εάν κάποιο μονοπάτι δεν ήταν σωστό, τότε έστω *v* η κορυφή με λανθασμένη απόσταση και τον μικρότερο αριθμό τοπολογικής διατάξεως
  – Τότε, εφ' όσον η απόσταση προς την *v* δύναται να μειωθεί, έστω ότι το αληθινά συντομότερο, για την *v*, μονοπάτι έχει τελευταία ακμή την (*x*,*v*)
  – Η *x* έχει αριθμό τοπολογικής διατάξεως μικρότερο από αυτόν της *v*. Άρα, το μονοπάτι από την *s* στην *x* έχει την σωστή απόσταση και η (*x*,*v*) δεν χρησιμοποιήθηκε για χαλάρωση (άτοπο)



Παράδειγμα ΣΜΜΠ σε ΚΑΓ



Ψευδοκώδικας

```
Algorithm dagSSSP(graph g, vertex s);
int[] topsort = topSortList(g, new int[g.V]); // O(V+E)
double[] dis = new double[g.V];
for (v = 0; v < g.V; v++){                  // # V φορές
  dis[v] = INFTY;
  g.T[v] = -1;
}
dis[s] = 0.0;
for (i = 0; i < g.V; i++)                   // χαλάρωση κατά τοπολογική διάταξη
  for (x = g.List[v = topsort[i]]; x != null; x = x.getNext()){
   if ((dis[v]+x.weight) < dis[w = x.v]){          // # E φορές
      dis[w] = dis[v]+x.weight;
       g.T[w] = v;
   }
```

- **Πολυπλοκότητα: $O(V+E)$**

Συντομότερα Μονοπάτια Όλων των Ζευγών

- Αιτεί:
  - για κάθε ζεύγος κορυφών *v*,*w*, το συντομότερο μονοπάτι που τα συνδέει, εάν υπάρχει
- Πρώτη Λύση:
  - Εκτέλεση *V* φορές του αλγορίθμου Dijkstra, με κόστος $O(VE\log V)$
  - Καλή, εφ' όσον:
    - δεν υπάρχουν αρνητικά βάρη, και
    - το γράφημα είναι αραιό

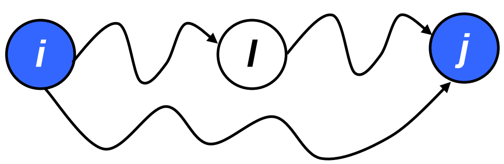
Αλγόριθμος Floyd

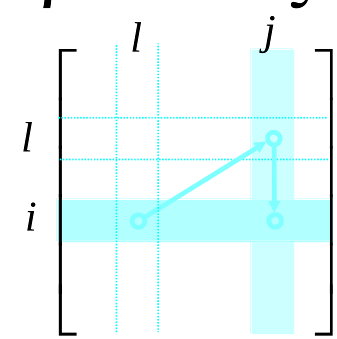
- Βασίζεται στην τεχνική του Δυναμικού Προγραμματισμού
- Ασφαλής επιλογή για πυκνά γραφήματα
- Ομοιάζει με τον Αλγόριθμο του Warshall για τον υπολογισμό της μεταβατικής κλειστότητας, ώστε αναφέρεται και ***ως Αλγόριθμος Floyd-Warshall***

Γενική Αρχή

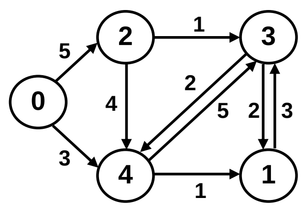
- Επιβάλλεται μία ***τυχαία*** διάταξη στις κορυφές και ακολουθούν *V* βήματα
- Κατά το *l*-στό βήμα, υπολογίζονται όλα τα συντομότερα μονοπάτια $\pi_{i,j}$ που χρησιμοποιούν, ως ενδιάμεσες κορυφές, ***μόνο*** τις *l* πρώτες, μέσω της χαλαρώσεως

$$d^{l}_{i,j} = \min\{d^{l-1}_{i,j}, d^{l-1}_{i,l} + d^{l-1}_{l,j}\}$$

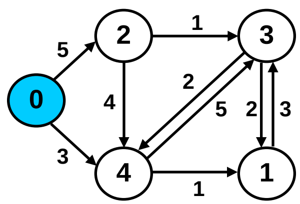




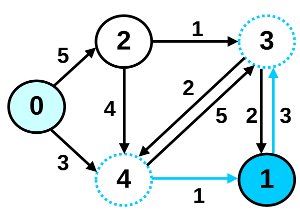
Παράδειγμα Τρεξίματος



$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & \infty & 5 & \infty & 3 \\ 1 & \infty & 0 & \infty & 3 & \infty \\ 2 & \infty & \infty & 0 & 1 & 4 \\ 3 & \infty & 2 & \infty & 0 & 2 \\ 4 & \infty & 1 & \infty & 5 & 0 \end{array} \qquad \begin{bmatrix} 0 & 5 & 2 & 5 & 4 \\ 5 & 1 & 5 & 3 & 5 \\ 5 & 5 & 2 & 3 & 4 \\ 5 & 1 & 5 & 3 & 4 \\ 5 & 1 & 5 & 3 & 4 \end{bmatrix}$$

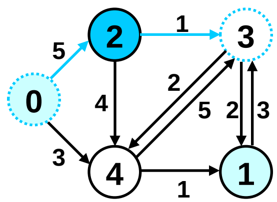


$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & \infty & 5 & \infty & 3 \\ 1 & \infty & 0 & \infty & 3 & \infty \\ 2 & \infty & \infty & 0 & 1 & 4 \\ 3 & \infty & 2 & \infty & 0 & 2 \\ 4 & \infty & 1 & \infty & 5 & 0 \end{array} \qquad \begin{bmatrix} 0 & 5 & 2 & 5 & 4 \\ 5 & 1 & 5 & 3 & 5 \\ 5 & 5 & 2 & 3 & 4 \\ 5 & 1 & 5 & 3 & 4 \\ 5 & 1 & 5 & 3 & 4 \end{bmatrix}$$

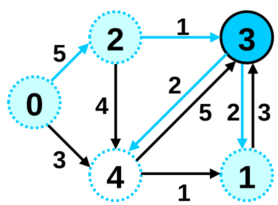


$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & \infty & 5 & \infty & 3 \\ 1 & \infty & 0 & \infty & 3 & \infty \\ 2 & \infty & \infty & 0 & 1 & 4 \\ 3 & \infty & 2 & \infty & 0 & 2 \\ 4 & \infty & 1 & \infty & 4 & 0 \end{array} \qquad \begin{bmatrix} 0 & 5 & 2 & 5 & 4 \\ 5 & 1 & 5 & 3 & 5 \\ 5 & 5 & 2 & 3 & 4 \\ 5 & 1 & 5 & 3 & 4 \\ 5 & 1 & 5 & 1 & 4 \end{bmatrix}$$

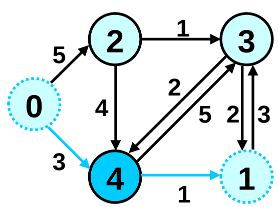
Παράδειγμα Τρεξίματος (συν.)



$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 0 & \infty & 5 & 6 & 3 \\
1 & \infty & 0 & \infty & 3 & \infty \\
2 & \infty & \infty & 0 & 1 & 4 \\
3 & \infty & 2 & \infty & 0 & 2 \\
4 & \infty & 1 & \infty & 4 & 0
\end{array}
\qquad
\begin{bmatrix}
0 & 5 & 2 & 2 & 4 \\
5 & 1 & 5 & 3 & 5 \\
5 & 5 & 2 & 3 & 4 \\
5 & 1 & 5 & 3 & 4 \\
5 & 1 & 5 & 1 & 4
\end{bmatrix}
$$



$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 0 & 8 & 5 & 6 & 3 \\
1 & \infty & 0 & \infty & 3 & 5 \\
2 & \infty & 3 & 0 & 1 & 3 \\
3 & \infty & 2 & \infty & 0 & 2 \\
4 & \infty & 1 & \infty & 4 & 0
\end{array}
\qquad
\begin{bmatrix}
0 & 2 & 2 & 2 & 4 \\
5 & 1 & 5 & 3 & 3 \\
5 & 3 & 2 & 3 & 3 \\
5 & 1 & 5 & 3 & 4 \\
5 & 1 & 5 & 1 & 4
\end{bmatrix}
$$



$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 0 & 4 & 5 & 6 & 3 \\
1 & \infty & 0 & \infty & 3 & 5 \\
2 & \infty & 3 & 0 & 1 & 3 \\
3 & \infty & 2 & \infty & 0 & 2 \\
4 & \infty & 1 & \infty & 4 & 0
\end{array}
\qquad
\begin{bmatrix}
0 & 4 & 2 & 2 & 4 \\
5 & 1 & 5 & 3 & 3 \\
5 & 3 & 2 & 3 & 3 \\
5 & 1 & 5 & 3 & 4 \\
5 & 1 & 5 & 1 & 4
\end{bmatrix}
$$

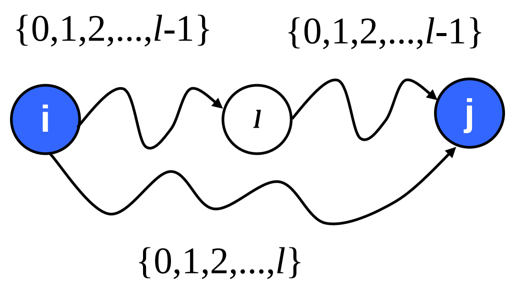
Ψευδοκώδικας

```
Algorithm FloydAPSP(graph g, double[][] d, int[][] path)
for (i = 0; i < g.V; i++)
  for (j = 0; j < g.V; j++){
     d[i][j] = INFTY;
     path[i][j] = g.V;
  }
for (i = 0; s < g.V; i++)
  for (j = 0; t < g.V; j++)
    if ((d[i][j] = g.A[i][j]) < INFTY)
      path[i][j] = j;
for (l = 0; l < g.V; l++)              // l-στο βήμα, #V φορές
  for (i = 0; i < g.V; i++)            // για κάθε i,  #V φορές
    if (d[i][l] < INFTY)
      for (j = 0; j < g.V; j++)        // και j, #V φορές
        if (d[i][j] > d[i][l]+d[l][j]){
            d[i][j] = d[i][l]+d[l][j];
            path[i][j] = path[i][l];   // καταγραφή της πρώτης κορυφής επί του π_{i,j}
        }
```

- **Πολυπλοκότητα: $O(V^3)$**

Ορθότητα

- Προκύπτει πολύ εύκολα από την αρχή της βέλτιστης υποδομής:
  – Κατά το *l*-στό βήμα, υπολογίζονται όλα τα συντομότερα μονοπάτια, αποτελούμενα ***αποκλειστικά*** από ενδιάμεσες κορυφές του συνόλου {0,1,2,...,*l*}



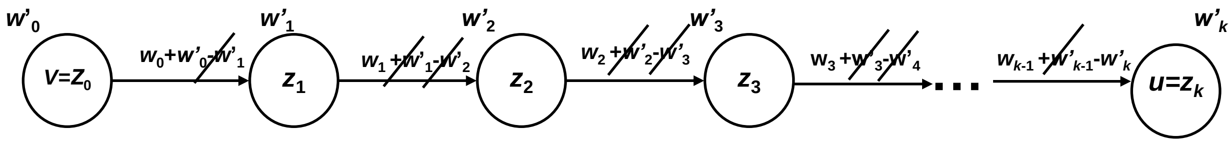
Συμπεράσματα

- Θετικά:
  - χειρίζεται και αρνητικά βάρη
  - εντοπίζει αρνητικούς κύκλους (στην διαγώνιο εμφανίζονται αρνητικές τιμές),
  - ασφαλή επιλογή για πυκνά γραφήματα:
    - εξοικονομείται ένας λογαριθμικός παράγοντας εν σχέσει με την επαναληπτική εφαρμογή Dijkstra
- Διαφορές με Warshall:
  - Άλλος ο στόχος (1 αντί για βάρος, $\vee$ αντί για min και $\wedge$ αντί για +)

Αλγόριθμος Johnson

- Η επιλογή Dijkstra συμφέρει για αραιά γραφήματα, όμως *δεν δουλεύει για αρνητικά βάρη*
- Johnson:
  - Παρατήρησε πως υπάρχει τρόπος εξαλείψεως αρνητικών βαρών, *εφ᾽ όσον δεν υπάρχουν αρνητικοί κύκλοι*

Μετασχηματισμός Johnson



- Τα βάρη όλων των μονοπατιών μεταξύ δύο κορυφών *v*, *u* μεταβάλλονται κατά την ίδια ποσότητα w'_0-w'_k:

$$w'_0+\sum(w_i+w'_i-w'_{i+1}) - w'_k$$

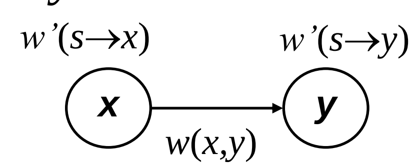
- Άρα, τα συντομότερα, παραμένουν συντομότερα…

Μετασχηματισμός (συν.)

- Για την εξάλειψη των αρνητικών βαρών, σε κάθε κορυφή v, ορίζεται -ως βάρος- η συντομότερη απόσταση από μία κορυφή-πηγή.
- Διακρίνονται δύο περιπτώσεις:
  - Η (x,y) ανήκει στο συντομότερο μονοπάτι $s \to y$.

    Άρα, η (x,y) είναι η τελευταία ακμή και $s \to x \to y = s \to y$. Συνεπώς,

    $$w(x,y) + w'(s \to x) - w'(s \to y) = w(x,y) - w(x,y) = 0$$

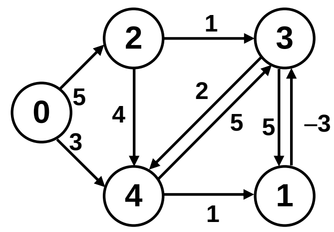


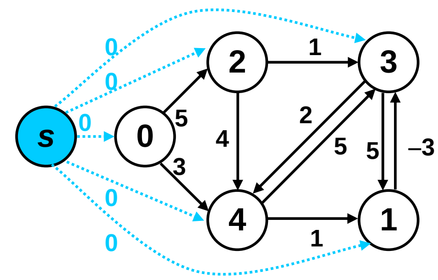
  - Η (x,y) δεν ανήκει στο συντομότερο μονοπάτι $s \to y$.

    Άρα, $s \to x \to y \neq s \to y$. Επομένως,

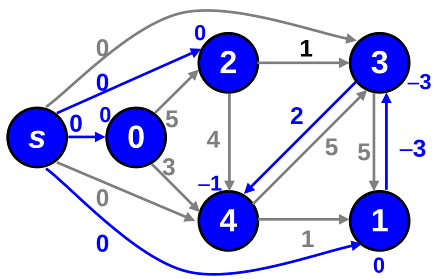
    $$w'(s \to x) + w(x,y) \geq w'(s \to y) \Rightarrow w(x,y) + w'(s \to x) - w'(s \to y) \geq 0$$

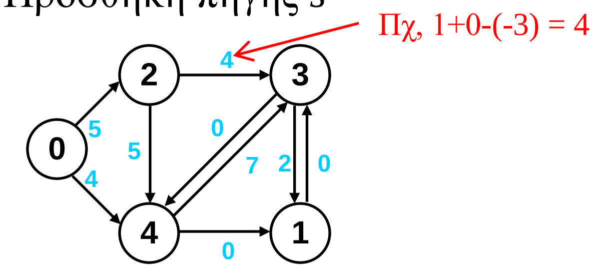
Παράδειγμα





(α) Προσθήκη πηγής *s*





(β) ΣΜΜΠ για την *s*

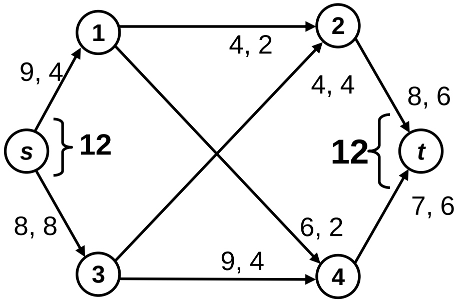
(γ) Μετασχηματισμός των βαρών

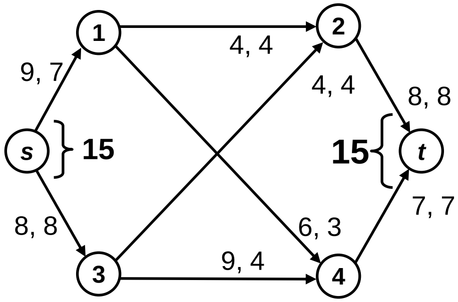
Αλγόριθμος Johnson

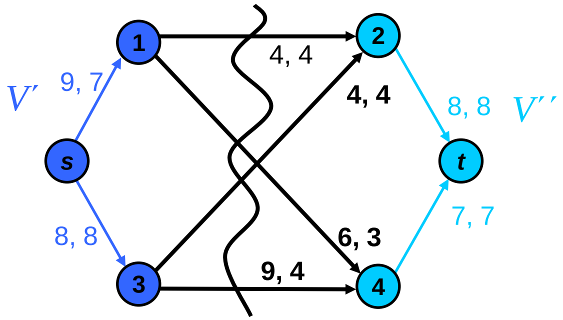
- Bellman-Ford για υπολογισμό ΣΜΜΠ ή εντοπισμό κύκλου- κόστος $O(VE)$
- Ο Bellman-Ford εφαρμόζεται από μία πηγή *s* που συνδέεται με μηδενικές ακμές.
  - Έτσι εξασφαλίζουμε ότι τα βάρη-αποστάσεις θα είναι πεπερασμένα
- V φορές Dijkstra- κόστος $O(VE\log V)=O(V^2\log V)$, εφ’όσον είναι αραιό

Ροές

- Ορισμοί
  - Δίκτυο Μεταφοράς ή Ροής
    - Βεβαρημένο γράφημα με χωρητικότητες
    - Πηγή, Καταβόθρα
  - Ροή
    - Απεικόνιση τιμών στις ακμές, φραγμένων από τις αντίστοιχες χωρητικότητες
    - Κορεσμένες, Ακόρεστες ακμές
    - Μέγιστη ροή
  - Τομή
    - Χωρητικότητα Τομής
    - Χωρητικότητα Δικτύου
- Ισοδυναμία Μέγιστης Ροής-Ελάχιστης Τομής

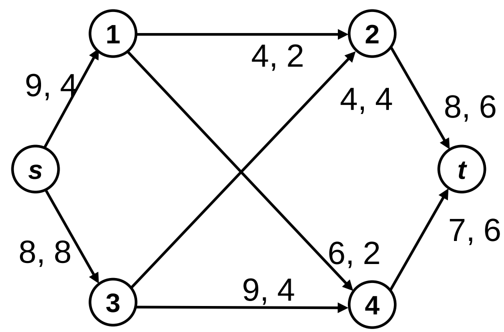


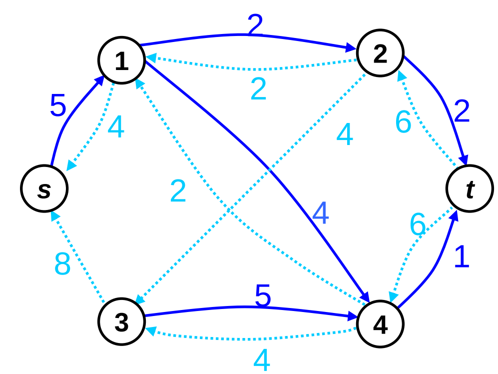


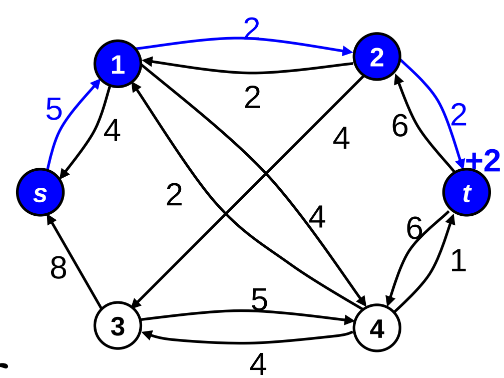


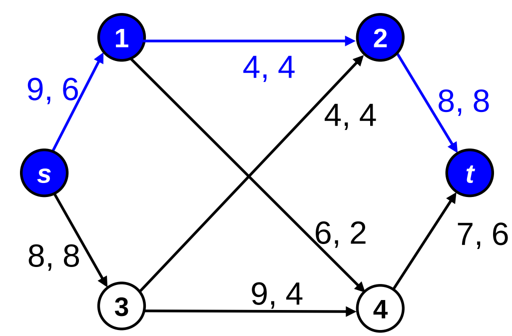
Βοηθητικές Συνδυαστικές Δομές

- Υπολειμματικό Γράφημα
  - Για κάθε ροή ακμής, περιέχει δύο ακμές: ***προωθήσεως*** και ***ακυρώσεως ροής***
- Επαυξητικό μονοπάτι
  - Κάθε μονοπάτι στο υπολειμματικό γράφημα από την πηγή στην καταβόθρα, με τιμή ροής ίση με την ελάχιστη τιμή ακμής
  - Βοηθά στην εύρεση, νέας, βελτιωμένης ροής









Θεώρημα Επαυξητικού Μονοπατιού

- Ροή Μέγιστη ανν το αντίστοιχο υπολειμματικό γράφημα δεν διαθέτει επαυξητικό μονοπάτι
  - Έστω μέγιστη ροή. Εάν υπήρχε επαυξητικό μονοπάτι, τότε θα ήταν δυνατή η επαύξηση (άτοπο)
  - Εάν δεν υπάρχει επαυξητικό μονοπάτι, οι κορυφές χωρίζονται σε δύο σύνολα *V*, *V’*: το πρώτο περιλαμβάνει τις προσπελάσιμες από την πηγή κορυφές και το δεύτερο τις υπόλοιπες. Τότε,
    - Ακμές *V* προς *V’* κορεσμένες
    - Ακμές *V’* προς *V* δίχως ροή
    - *V*, *V’* τομή, όπου η χωρητικότητα φράσσει την ροή

Θεώρημα Ακέραιας Ροής

- Εάν οι χωρητικότητες είναι ακέραιες τότε και η μέγιστη ροή έχει ακέραια τιμή
  - Παρατηρούμε ότι τα επαυξητικά μονοπάτια έχουν ακέραιες τιμές ροών

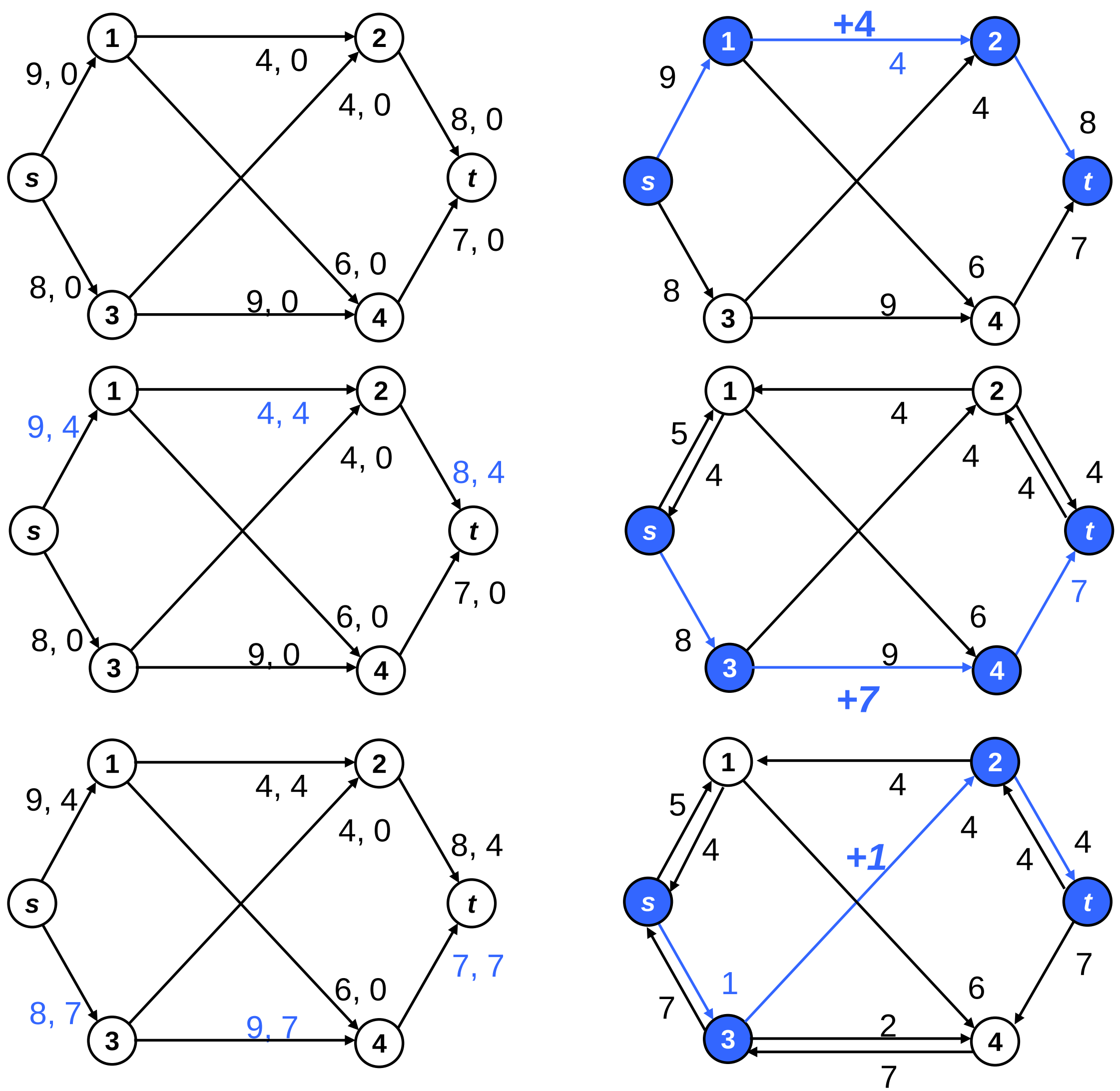
Μέθοδος Ford-Fulkerson

- Βασίζεται στο θεώρημα επαυξητικού μονοπατιού:

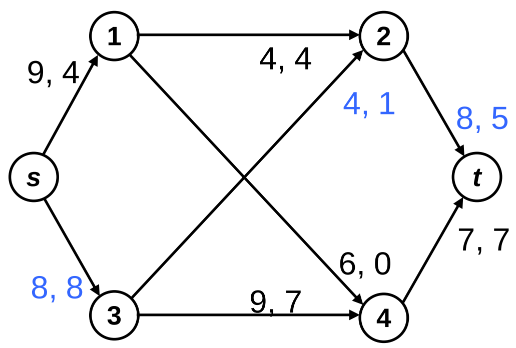
  «*όσο υπάρχουν επαυξητικά μονοπάτια, διαλέγουμε ένα και επαυξάνουμε την ροή κατά μήκος του*»

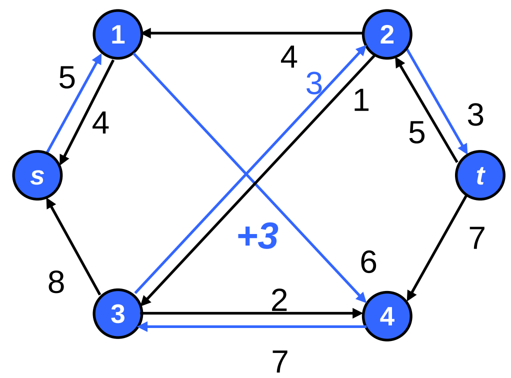
- Αποκαλείται μέθοδος, γιατί ***δεν*** προσδιορίζεται πώς θα βρεθεί το επαυξητικό μονοπάτι
- Στην σχετική δημοσίευση, οι Ford-Fulkerson είχαν προτείνει γενικευμένο ψάξιμο στο υπολειμματικό γράφημα, με ***τυχαία*** ουρά προτεραιότητας

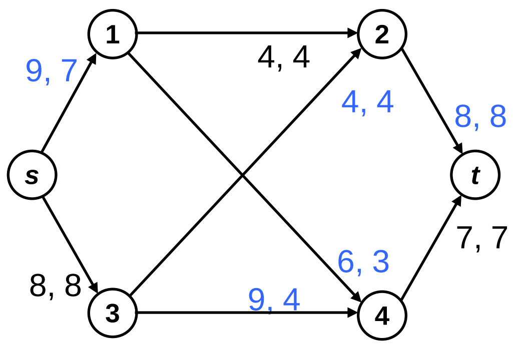
Παράδειγμα

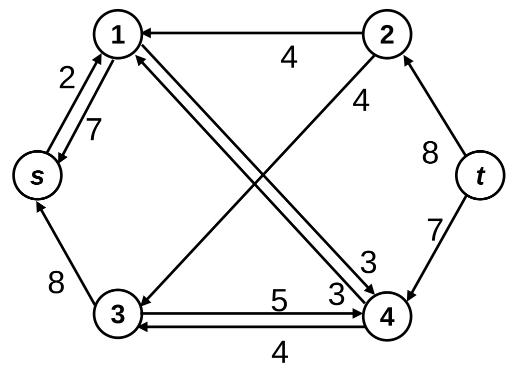

1
2
s
t
3
4
9, 0
4, 0
4, 0
8, 0
8, 0
7, 0
6, 0
9, 0
+4
4
9
4
8
8
7
6
9
9, 4
4, 4
4, 0
8, 4
8, 0
7, 0
6, 0
9, 0
5
4
4
4
4
4
8
9
6
7
+7
9, 4
4, 4
4, 0
8, 4
8, 7
7, 7
6, 0
9, 7
5
4
4
+1
4
4
4
1
7
7
6
2
7

Παράδειγμα (συν.)



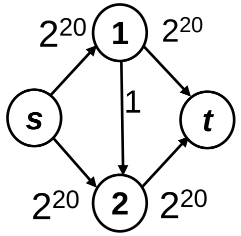






Ανάλυση Μεθόδου

- Πολυπλοκότητα $O(VEM)$=O(|f|E)
  - Στην χειρότερη περίπτωση, κάθε βήμα θα αυξάνει την χωρητικότητα κατά μία μονάδα.
  - Άρα |f | ψαξίματα μονοπατιού, κόστους $O(V+E)$=O(E).
  - Επιπλέον, |f |=$O(VM)$ καθώς η τιμή τομής είναι $O(VM)$
  - Το όριο είναι ***σφιχτό*** (**tight**), όπως δείχνει και το διπλανό στιγμιότυπο, και ***ψευδοπολυωνυμικό***, γιατί εξαρτάται από την |f |



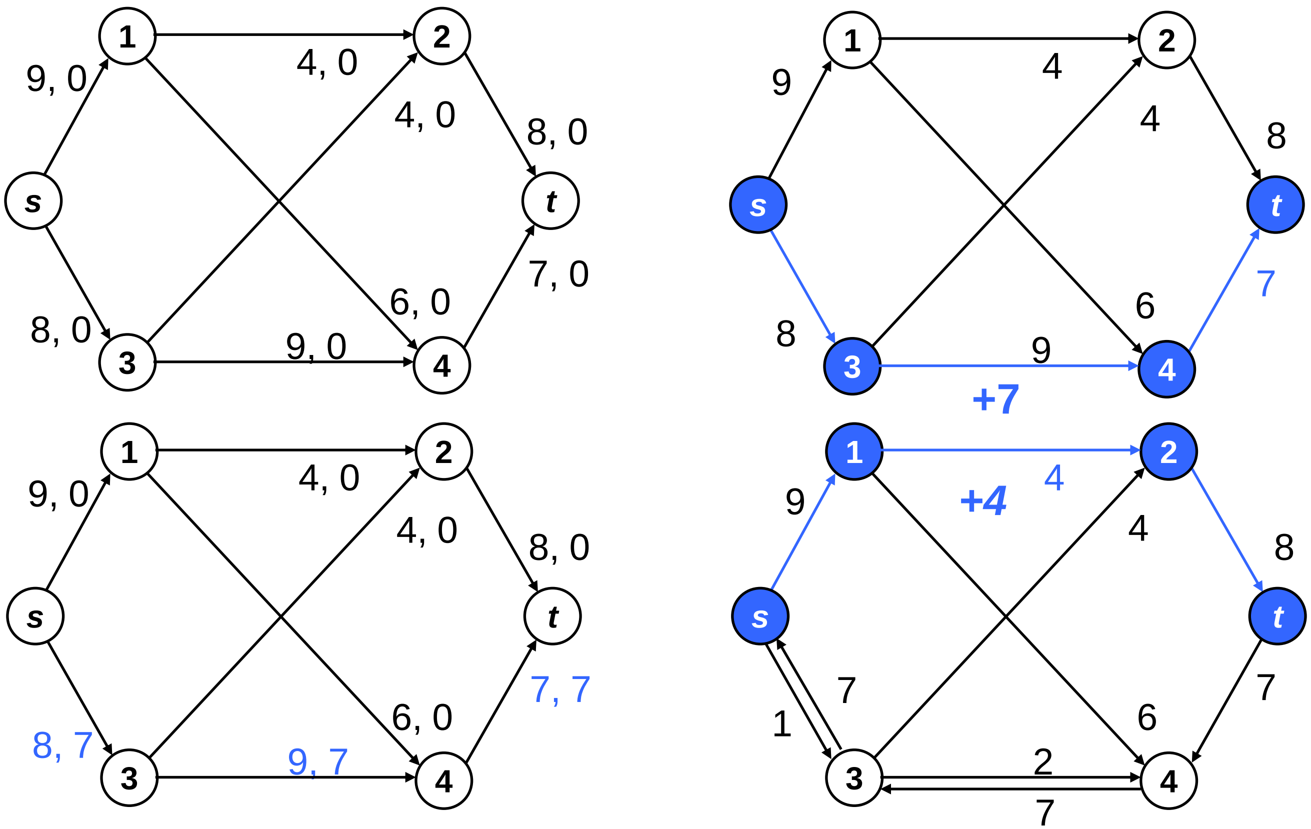
- Άλλο μειονέκτημα:
  - Αστάθεια συγκλίσεως στην τελική ροή, εάν οι χωρητικότητες είναι πραγματικοί αριθμοί

Αλγόριθμοι Edmonds-Karp

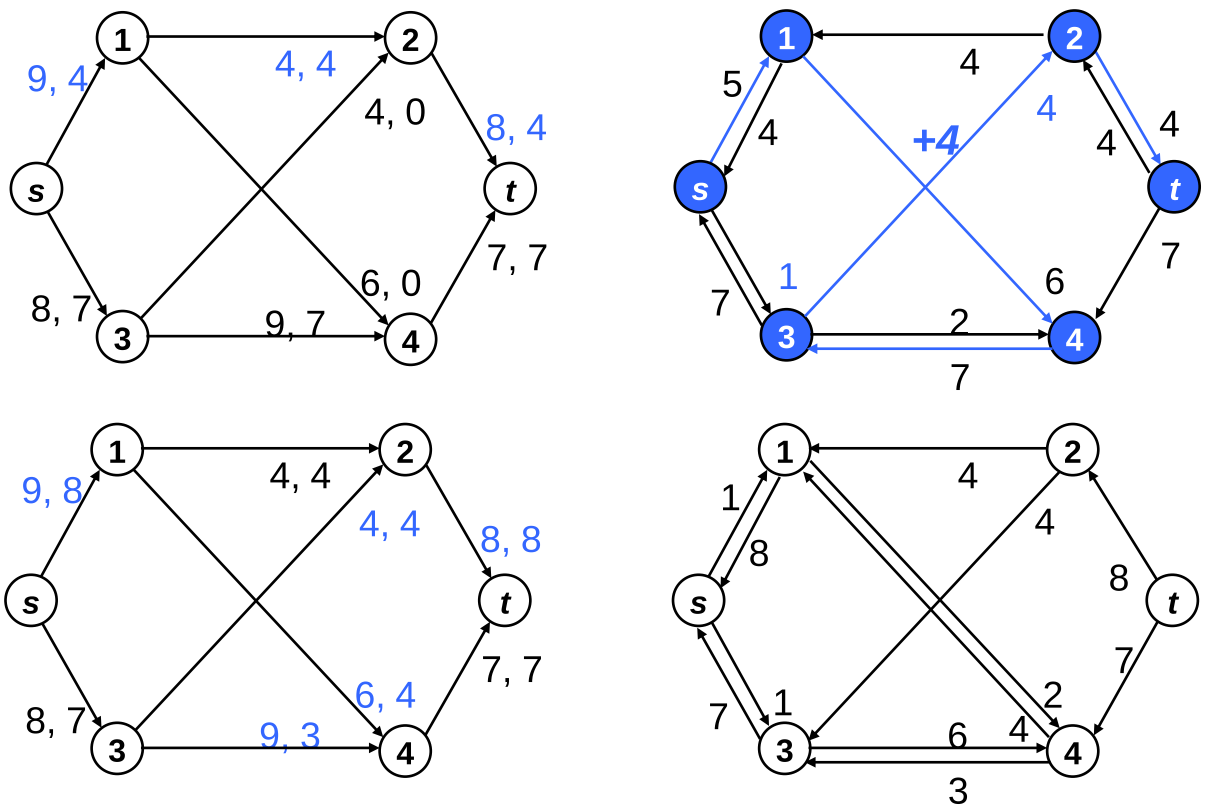
- Οι Edmonds-Karp παρατήρησαν πως, εάν τα επαυξητικά μονοπάτια αναζητηθούν
  - είτε *βάσει της μέγιστης τιμής ροής*
  - είτε *βάσει του μικρότερου πλήθους ακμών*,

  τότε η Μέθοδος Ford-Fulkerson γίνεται ***πολυωνυμική***

Παράδειγμα Μέγιστης Τιμής Επαυξητικού Μονοπατιού



Παράδειγμα Μέγιστης Τιμής Επαυξητικού Μονοπατιού (συν.)



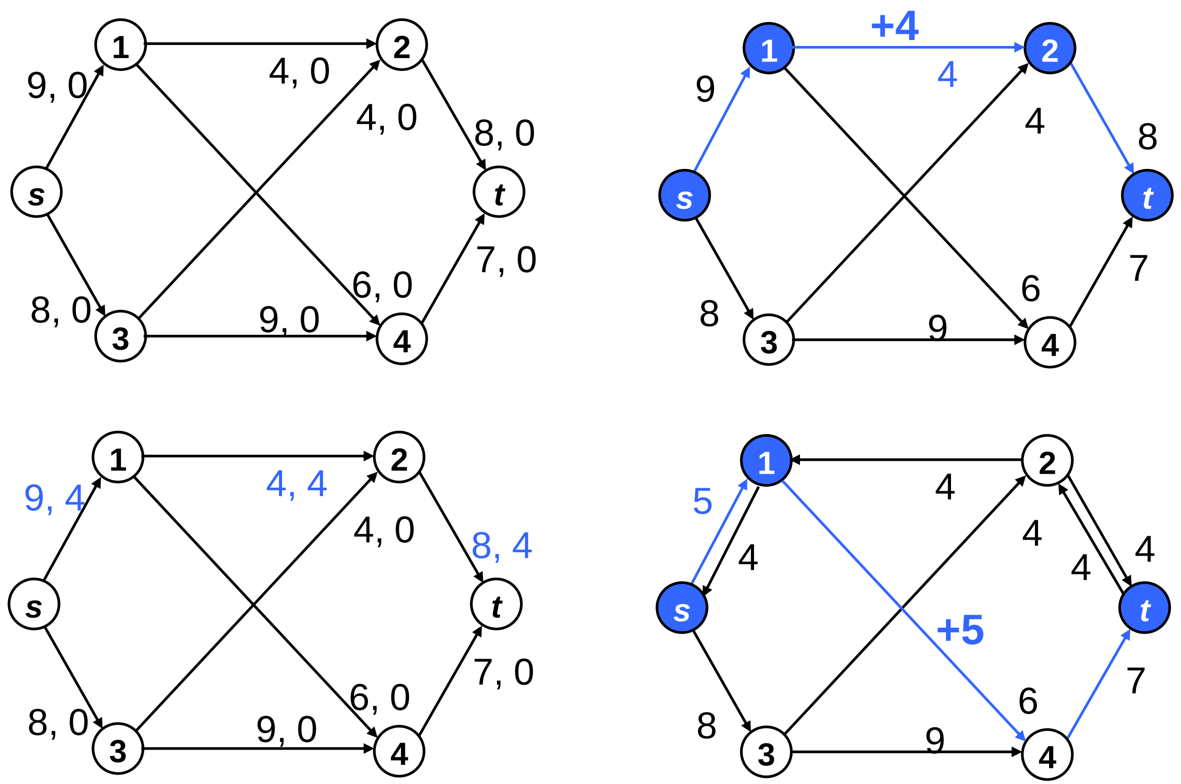
Ανάλυση

- Το πλήθος των επαυξητικών μονοπατιών είναι $O(E\log M)$, M μέγιστη τιμή χωρητικότητας
  - Έστω ότι η ροή f' έχει τιμή $|f'|$ και $|f|$ είναι η τελική τιμή
  - Υπάρχουν το πολύ E διακριτά επαυξητικά μονοπάτια, συνολικής ροής $|f|-|f'|$
  - Βάσει της αρχής του περιστερώνα, το μέγιστο μονοπάτι θα έχει τιμή ***τουλάχιστον*** $(|f|-|f'|)/E$
  - Θεωρούμε την ακολουθία των επαυξητικών βημάτων σε υποομάδες των 2Ε. Έστω μία υποομάδα $2E$ επαυξητικών βημάτων που δημιουργούν ροές $f'_1, f'_2, f'_3, \ldots, f'_i, \ldots$ και f' η ροή πριν την έναρξή της. Τότε,
    - Εάν κάθε ένα από αυτά τα βήματα προωθεί ροή τιμής *τουλάχιστον* $|f|-|f'|/2E$, τότε η διαδικασία τερματίζει μετά από $2E$ βήματα, το πολύ, και αυτή είναι η τελευταία υποομάδα που μελετούμε. Διαφορετικά,
    - Εάν υπάρχει έστω και ένα επαυξητικό βήμα, πχ, το i-στο, με τιμή ροής λιγότερη από $(|f|-|f'|)/(2E)$, τότε

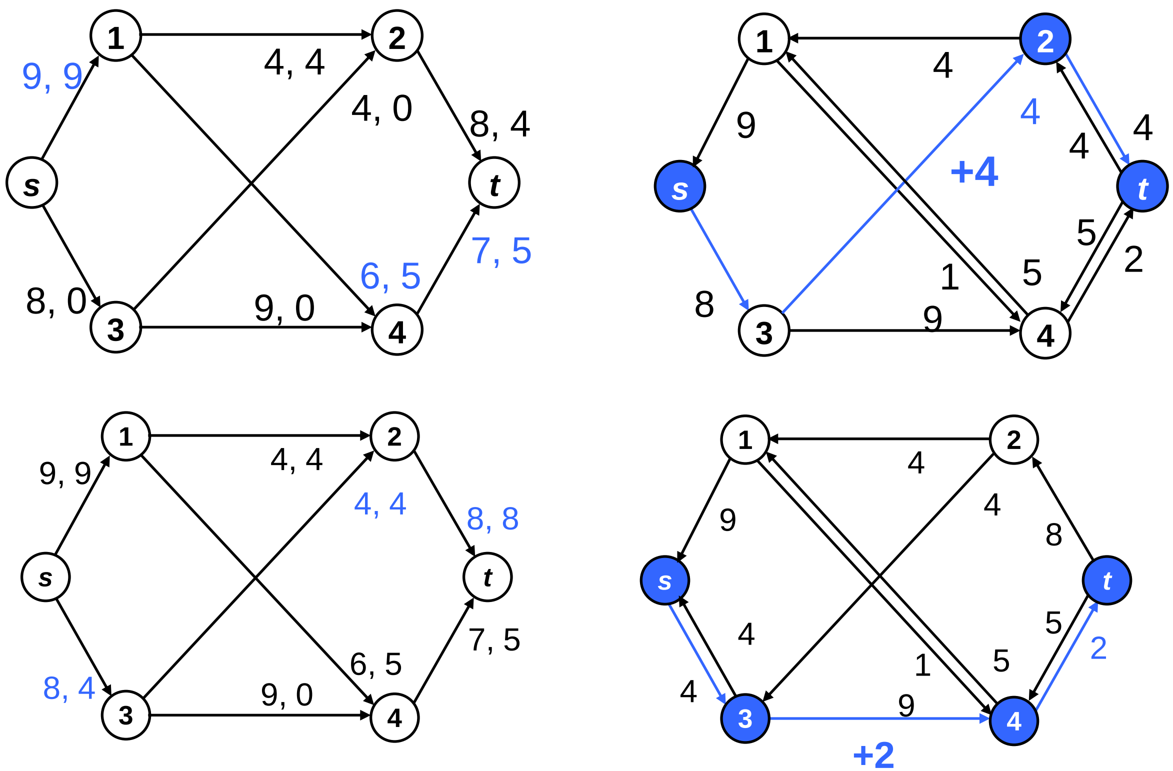
    $$(|f|-|f'_{i-1}|)/E \le |f'_i|-|f'_{i-1}| < (|f|-|f'|)/(2E) \Rightarrow [(|f|-|f'_{i-1}|)/E\,]/[\,(|f|-|f'|)/E] \le 1/2$$

    - Άρα, στην χειρότερη περίπτωση, μετά από $2E$ βήματα, υποδιπλασιάζεται η τιμή του «βαρύτερου» μονοπατιού
- Χρόνος $O(\log_{E/V}V \log M\, E(V+E))$, M μέγιστη χωρητικότητα ακμής
  - $V+E$ πράξεις στην ουρά προτεραιότητας, πχ, E/V-σωρός, σε κάθε έναν από τους $E\log M$ υπολογισμούς επαυξητικού μονοπατιού

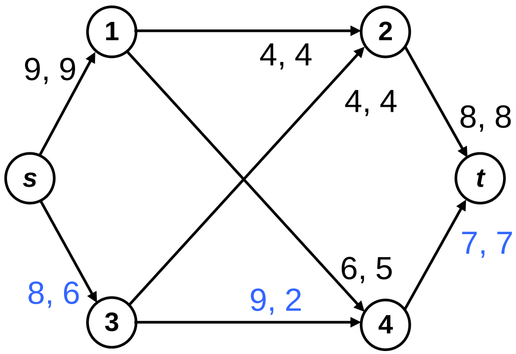
Παράδειγμα Συντομότερου Επαυξητικού Μονοπατιού

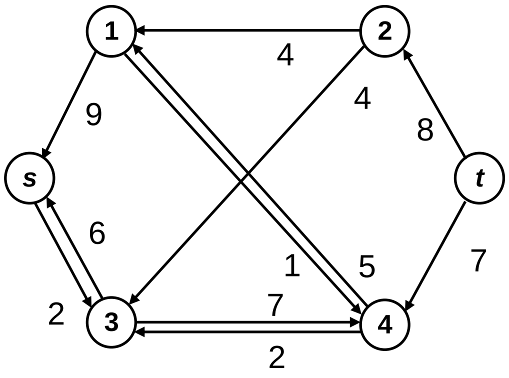


Παράδειγμα Συντομότερου Επαυξητικού Μονοπατιού (συν.)



Παράδειγμα Συντομότερου Επαυξητικού Μονοπατιού (συν.)

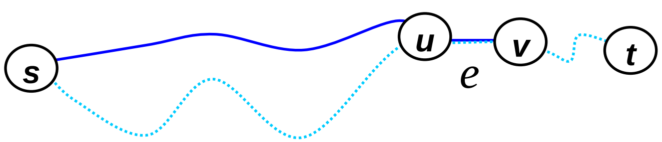




Ανάλυση

- **Ορθότητα (Σχεδιάγραμμα)** Στηρίζεται στις κάτωθι παρατηρήσεις:
  - **Πρώτον,** τα συντομότερα μονοπάτια συνεχώς αυξάνουν σε μήκος, και
  - **Δεύτερον,** κάθε ακμή του υπολειμματικού γραφήματος ξαναεμφανίζεται σε συντομότερο μονοπάτι που είναι κατά 2 ακμές μεγαλύτερο από την προηγούμενη φορά:

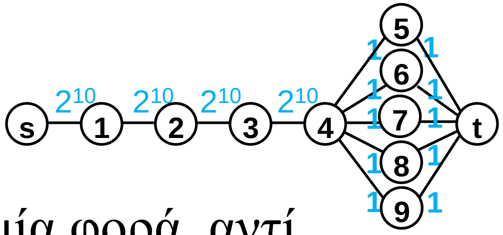
$$d(u) = d(v) + 1 \geq d'(v) + 1 = d'(u) + 1 + 1 = d'(u) + 2$$



  - Άρα, θα χρησιμοποιηθούν συνολικά $VE/2$ επαυξητικά μονοπάτια
- Χρόνος $O(E(VE))=O(VE^2)$

Τεχνική Προροής-Προωθήσεως

- Διαίσθηση
  - Γιατί να μην σταλεί με μιας ροή 5 στην κορυφή 4;
  - Το $s \rightarrow 4$ θα συμμετείχε μόνο μία φορά, αντί σε 5 διαφορετικά επαυξητικά μονοπάτια, μεγέθους 1



- Τεχνική Προροής-Προωθήσεως
  - Προσπαθούμε να οδηγήσουμε σε κορεσμό τις ακμές, διοχετεύοντας όση ροή επιτρέπεται
  - Οι ενδιάμεσες κορυφές μπορούν να αποθηκεύουν πλεονάζουσα ροή – χαρακτηρίζονται, τότε, ως ***ενεργές (active)***
  - Έτσι δημιουργείται μία προσέγγιση ροής, η ***προροή***, η οποία θα συγκλίνει στην τελική «κανονική» ροή

Τεχνική Προροής-Προωθήσεως (συν.)

- Προκειμένου η προροή να μετασχηματιστεί σε νόμιμη ροή, πρέπει πρώτα να φθάσει ροή στην καταβόθρα και, κατόπιν, το πλεόνασμα να επιστραφεί στην πηγή
- Δηλαδή, ***πρέπει να εξαλειφθούν οι ενεργές κορυφές***. Πώς;
  – Υπολειμματικό γράφημα
  – Συνάρτηση ύψους στο υπολειμματικό γράφημα
- Συνάρτηση ύψους
  – $h(t) = 0$
  – $h(v) \leq h(u) + 1$, $(v,u) \in E_r$

  Π.χ., $h \equiv$ απόσταση από καταβόθρα
- Όταν $h(v) = h(u) + 1$, τότε η (v,u) καλείται ***νόμιμη***

Ιδιότητες

- Το ύψος μίας κορυφής v δεν ξεπερνά το μήκος της συντομότερης αποστάσεως από την v προς την καταβόθρα t
  - Έστω $v \to v_1 \to v_2 \to \cdots \to t$ το συντομότερο μονοπάτι. Τότε:

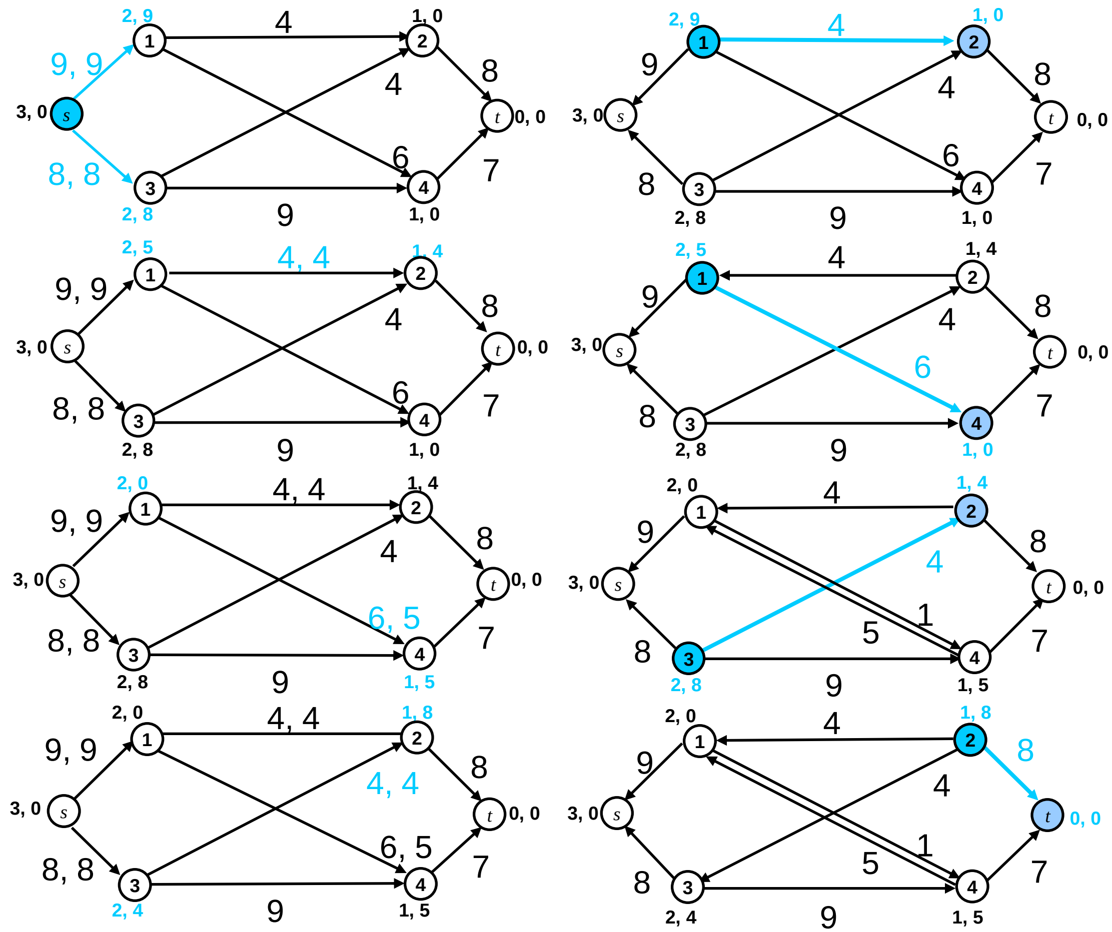
$$h(v) \le h(v_1) + 1 \le h(v_2) + 2 \le \cdots \le h(t) + d = 0 + d$$

- Όταν $h(v) > V$, τότε δεν υπάρχει μονοπάτι από την v προς την καταβόθρα t (γιατί;)

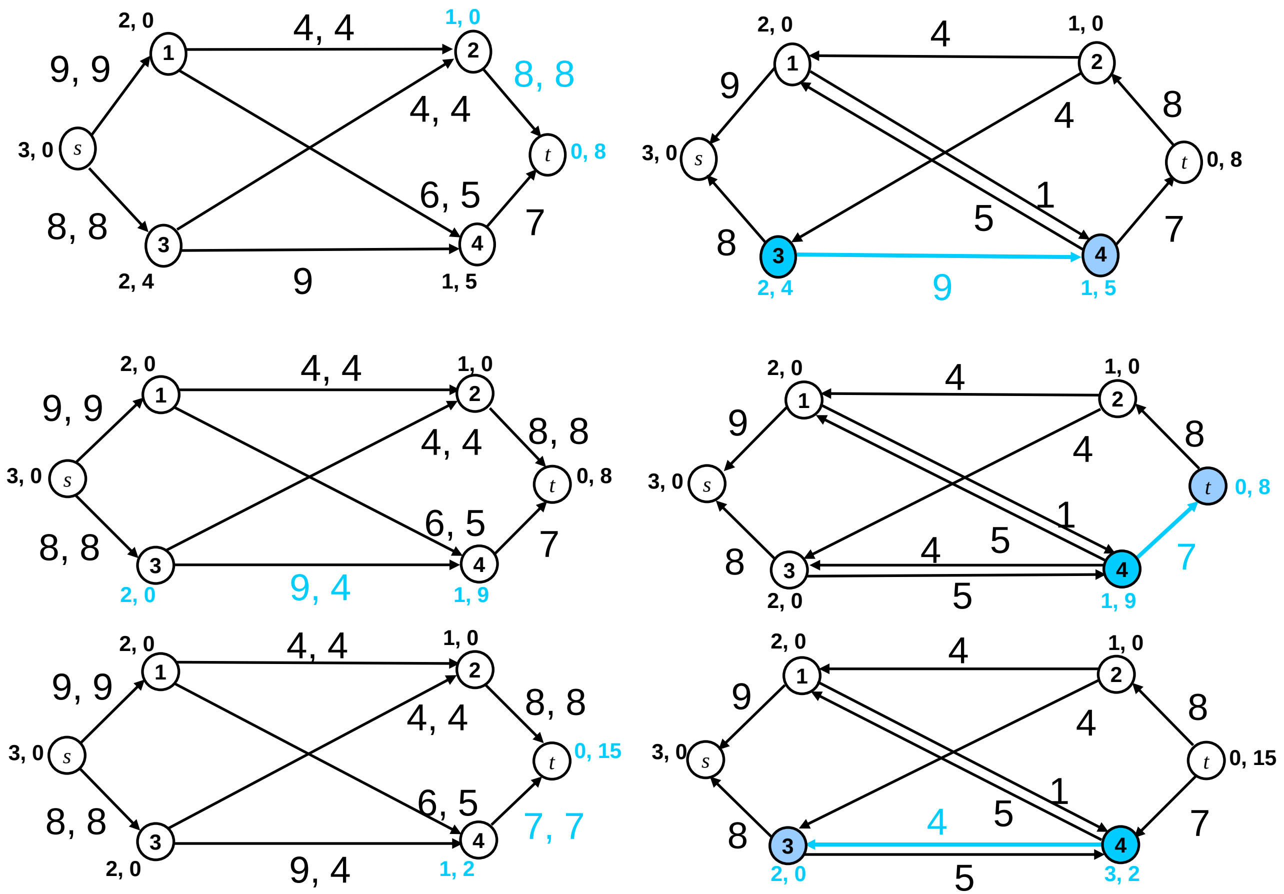
Γενικό Σχεδιάγραμμα Αλγορίθμων Προροής-Προωθήσεως

- Αναθέτουμε ύψη βάσει των συντομότερων μονοπατιών προς καταβόθρα
- Φέρουμε σε κορεσμό όλες τις ακμές της πηγής
- Όσο υπάρχουν ενεργές κορυφές, επιλέγουμε μία, έστω v, και
  - είτε σπρώχνουμε ροή μέσω μίας ***νόμιμης ακμής*** της
  - είτε, εάν δεν υπάρχουν νόμιμες ακμές, υψώνουμε την v, ώστε να είναι κατά ***μία μονάδα υψηλότερη*** από τον ***χαμηλότερό της γείτονα***

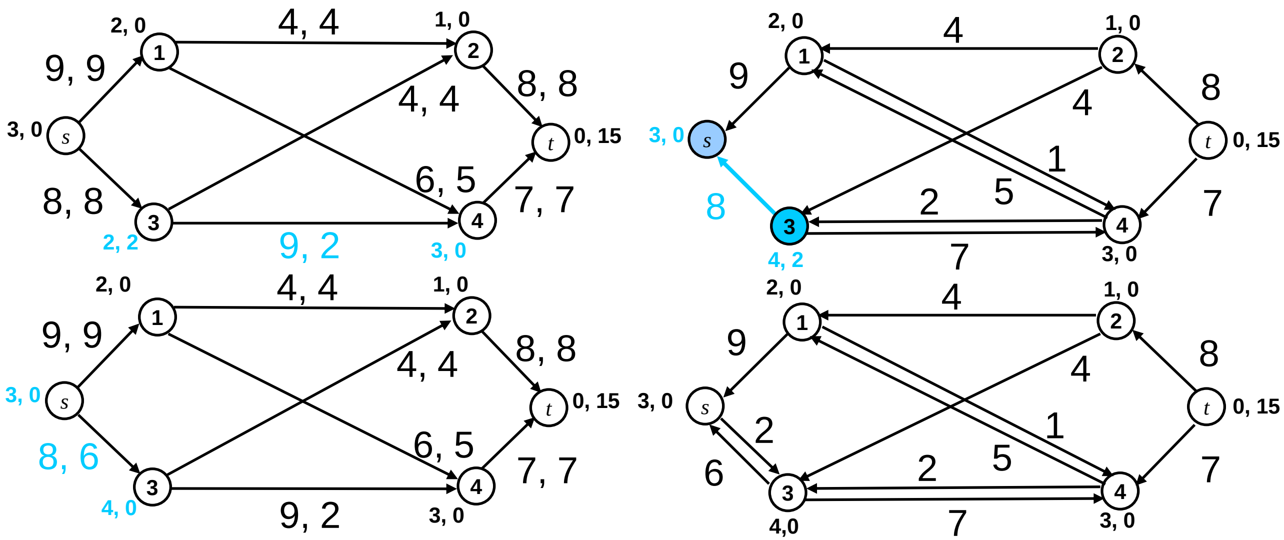
Παράδειγμα


2, 9
4
1, 0
1
2
9, 9
8
4
3, 0
s
t
0, 0
6
8, 8
3
4
7
2, 8
9
1, 0
2, 9
4
1, 0
1
2
9
8
4
3, 0
s
t
0, 0
6
8
3
4
7
2, 8
9
1, 0
2, 5
4, 4
1, 4
1
2
9, 9
8
4
3, 0
s
t
0, 0
6
8, 8
3
4
7
2, 8
9
1, 0
2, 5
4
1, 4
1
2
9
8
4
3, 0
s
t
0, 0
6
8
3
4
7
2, 8
9
1, 0
2, 0
4, 4
1, 4
1
2
9, 9
8
4
3, 0
s
t
0, 0
6, 5
8, 8
3
4
7
2, 8
9
1, 5
2, 0
4
1, 4
1
2
9
8
4
3, 0
s
t
0, 0
1
5
8
3
4
7
2, 8
9
1, 5
2, 0
4, 4
1, 8
1
2
9, 9
8
4, 4
3, 0
s
t
0, 0
6, 5
8, 8
3
4
7
2, 4
9
1, 5
2, 0
4
1, 8
1
2
8
9
4
3, 0
s
t
0, 0
1
5
8
3
4
7
2, 4
9
1, 5

Παράδειγμα (συν.)



Παράδειγμα (συν.)



Γιατί δουλεύει;

- Μηχανικό ανάλογο
- Στο υπολειμματικό γράφημα υπάρχει μονοπάτι από κάθε ενεργή κορυφή προς την πηγή, ενώ η πηγή είναι απομονωμένη από την καταβόθρα
  - *Πολύ εύκολα, με επαγωγή, ξεκινώντας από το γεγονός πως, αρχικά, οδηγούνται σε κορεσμό αυτές της πηγής* s
  - *Οι μόνες που «βλέπει» η πηγή, έχουν ύψος >* $h(s)$ *(αφού της επιστρέφουν ροή)* $\Rightarrow$ *δεν επικοινωνούν με την καταβόθρα (ιδιότητα ύψους* $h(v) \leq h(u) + 1$, $(v,u) \in E_r$, *σταθερό 0)*
- Καμμία κορυφή δεν αποκτά ύψος μεγαλύτερο του $2V$
  - *Η πηγή έχει ύψος* $\leq V$, *ενώ κάθε ενεργή κορυφή v απέχει το πολύ* V-2 *ακμές από την* s: $v \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow s$

    $$h(v) \leq h(v_1) + 1 \leq h(v_2) + 2 \leq \cdots \leq h(s) + d \leq V + d \leq 2V\text{-}2$$
  - *Όταν η v καταστεί ανενεργή, το ύψος της είτε είναι το ίδιο είτε κατά ένα μεγαλύτερο από τον χαμηλότερο γείτονά της.*

Γιατί δουλεύει; (συν.)

- Όταν τερματίζει:
  - δεν υπάρχουν ενεργές κορυφές
  - η πηγή είναι αποκομμένη από την καταβόθρα
- Άρα
  - δημιουργείται κανονική ροή
  - δεν υπάρχουν επαυξητικά μονοπάτια

Ανάλυση

- Μέσω Συναρτήσεων Δυναμικού (σχεδιάγραμμα)
  - $\mathbf{\Phi} = \sum h(v)$, v ενεργή
    - αρχική τιμή $\leq V \times 2V$
    - τελική τιμή 0
  - Πώς μεταβάλλεται το δυναμικό
    - Α΄ περίπτωση: ύψωση κορυφής κατά $c \Rightarrow \Delta\mathbf{\Phi} = c$. Συνολικά, $2V^2$ $(V \times 2V)$
    - Β΄ περίπτωση: διοχέτευση ροής.
      - Κορεσμός ακμής:
        $\Delta\mathbf{\Phi} = \mathbf{2}V \Rightarrow 2V^2 E$ (τυχόν ενεργοποίηση κορυφής απολήξεως)
      - Ακόρεστη ακμή:
        $\Delta\mathbf{\Phi} = -h(v)+h(v)-1 = -1$ (γίνεται ενεργή γειτονική w, με $h(w)= h(v)-1$) ή
        $\Delta\mathbf{\Phi} = -h(v)$ (ήδη ενεργή η γειτονική w)

Ανάλυση (συν.)

- Σύνοψη
  - Αρχικά:

    Φ $= 2V^2$
  - Μεταβολές:

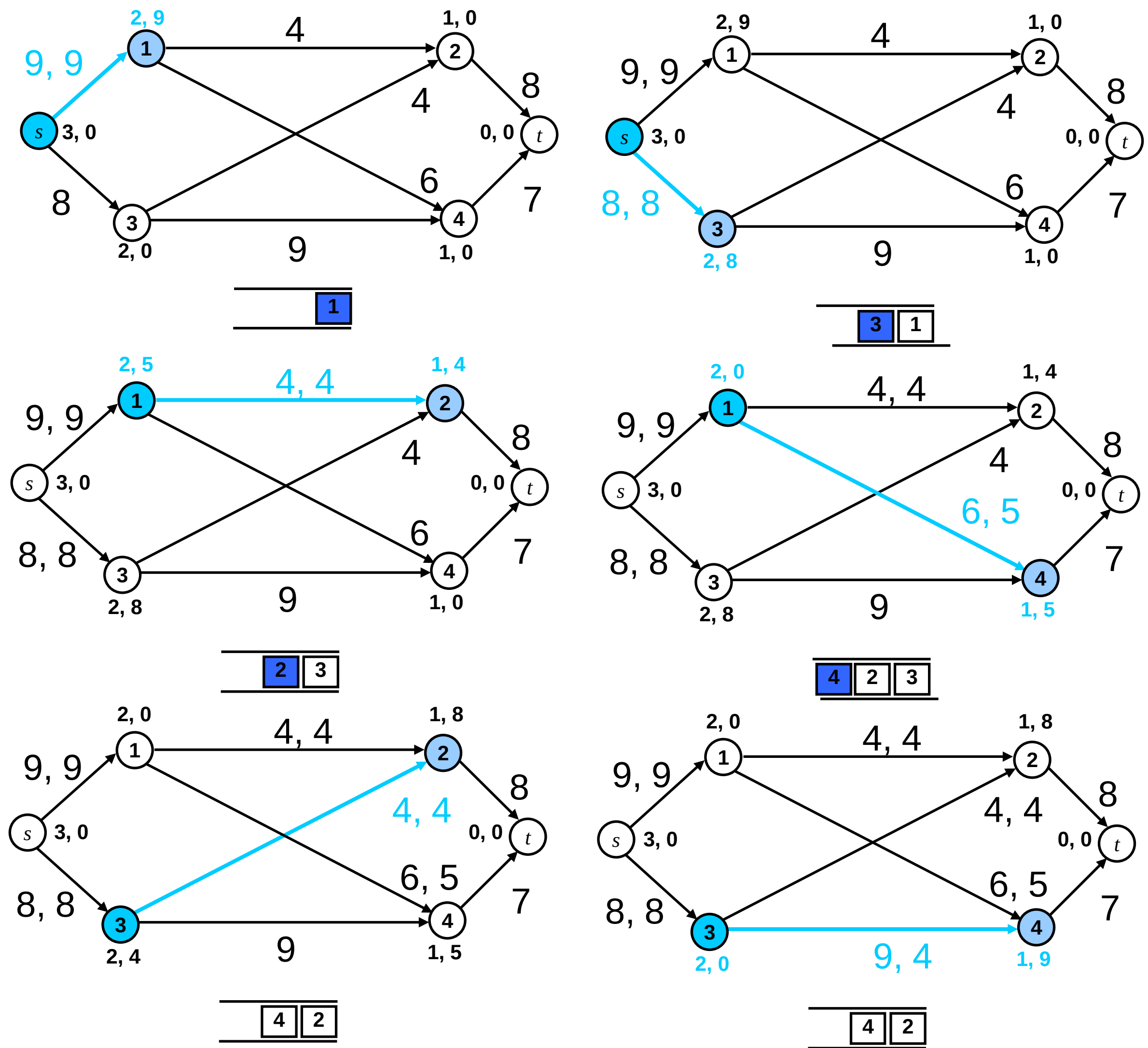
    $+(2V^2+2\,V^2E)$ από αλλαγές ύψους/κορεσμούς ακμών
  - Τελικά:

    Φ $= 0 \Rightarrow 2V^2+2\,V^2E = O(V^2E)$ ακόρεστες προωθήσεις
    (έκαστη προκαλεί Δ**Φ** ≤ -1)
- Χρόνος $O(V^2E)$

Τροποποίηση Goldberg-Tarjan

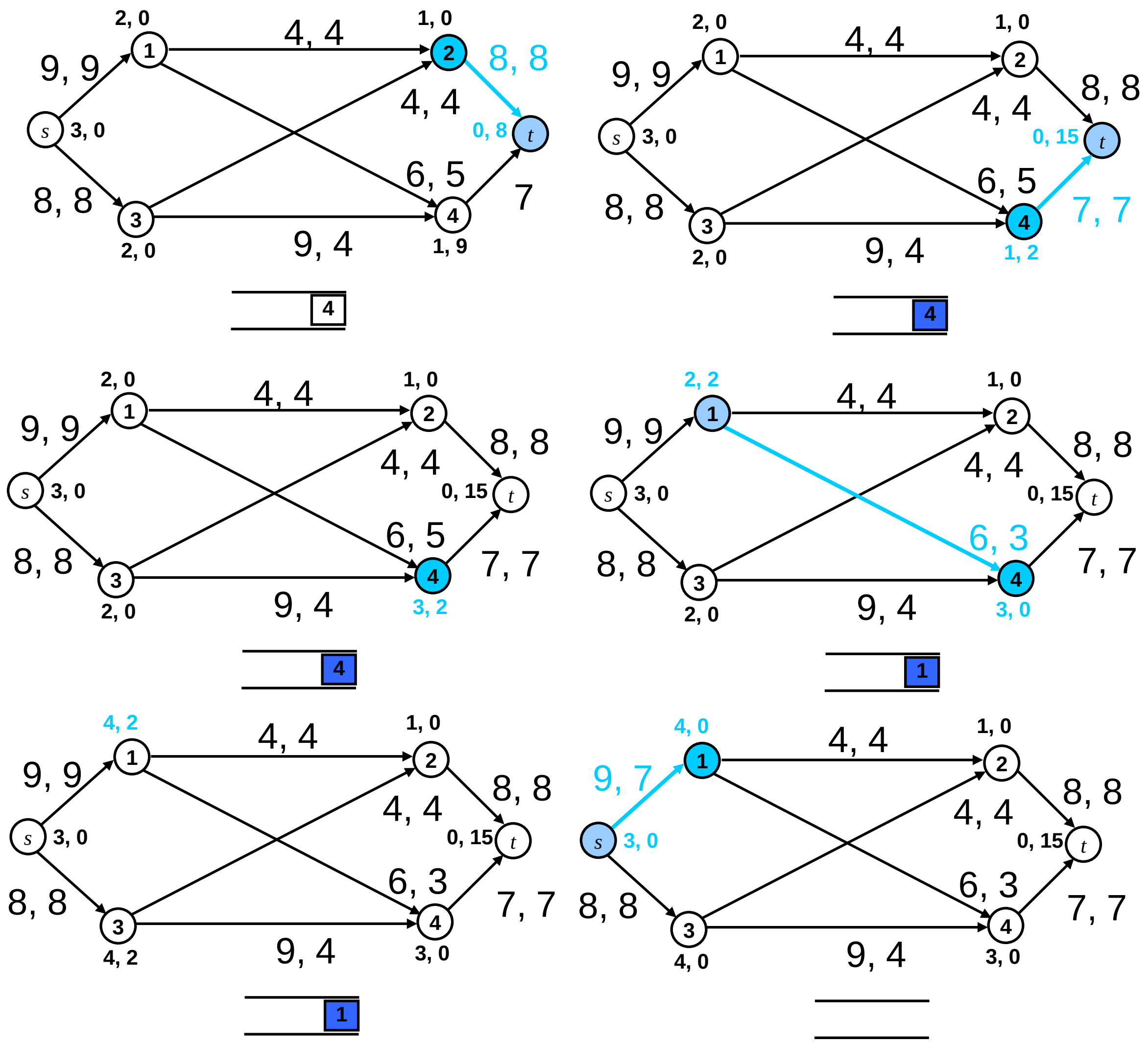
- Επιλογή ενεργής κορυφής για προώθηση ροής μέσω ουράς FiFo
- Όταν επιλεγεί ενεργή κορυφή, προωθείται ***συνεχώς*** ροή μέσω νομίμων ακμών μέχρι
  - είτε να καταστεί ανενεργή
  - είτε να τελειώσουν οι νόμιμες ακμές, οπότε προσαρμόζεται το ύψος της

Παράδειγμα



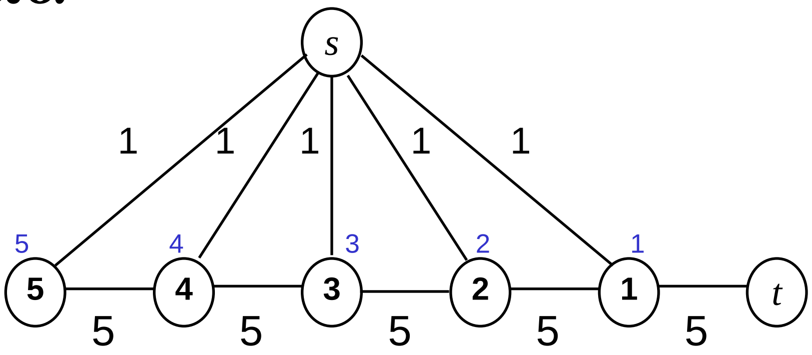
2, 9
1, 0
9, 9
4
1
2
8
4
s
3, 0
0, 0
t
6
7
8
3
4
2, 0
9
1, 0
1
2, 9
1, 0
9, 9
4
8
4
3, 0
0, 0
6
7
8, 8
9
2, 8
1, 0
3
1
2, 5
1, 4
9, 9
4, 4
8
4
3, 0
0, 0
6
8, 8
7
2, 8
9
1, 0
2
3
2, 0
1, 4
9, 9
4, 4
8
4
3, 0
0, 0
6, 5
8, 8
7
9
2, 8
1, 5
4
2
3
2, 0
1, 8
9, 9
4, 4
8
4, 4
3, 0
0, 0
6, 5
8, 8
7
2, 4
9
1, 5
4
2
2, 0
1, 8
9, 9
4, 4
8
4, 4
3, 0
0, 0
6, 5
8, 8
7
2, 0
9, 4
1, 9
4
2

Παράδειγμα (συν.)



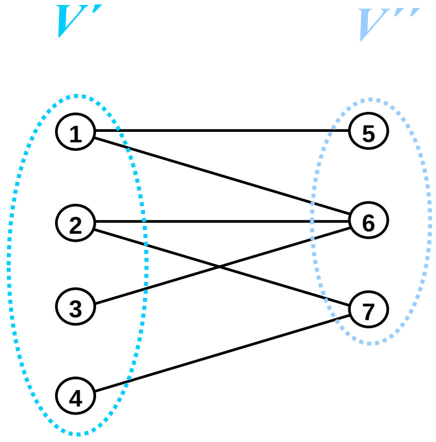
Ανάλυση

- Πολυπλοκότητα $O(V^2E)$
  - Αποδεικνύεται με συνάρτηση δυναμικού
- Το όριο είναι σφικτό, όπως δείχνει και το κάτωθι σχήμα



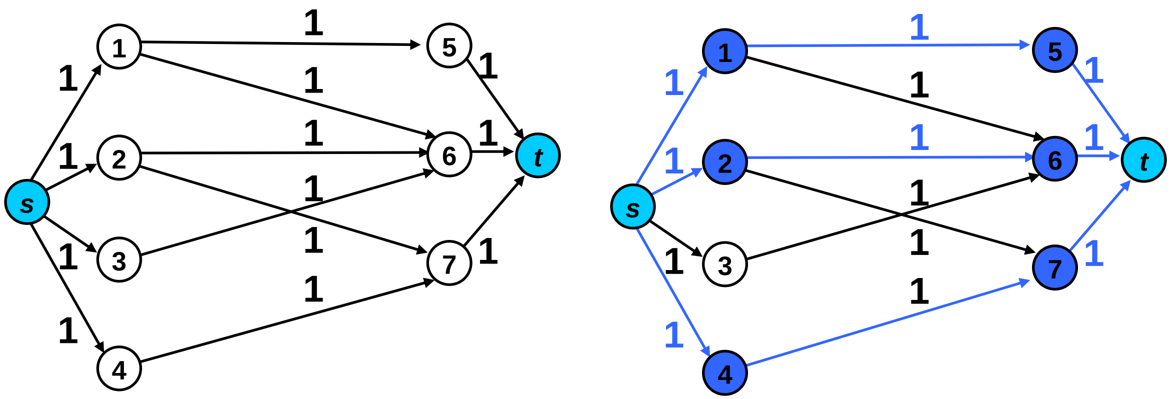
Μέγιστα Ταιριάσματα σε Διμελή Γραφήματα

- Ορισμοί
  - Κάθε σύνολο ακμών με διαζευγμένα άκρα
  - Μέγιστο, εάν δεν υπάρχει άλλο πολυπληθέστερο που να το περιέχει



Επίλυση

- Αναγωγή σε πρόβλημα ροής
  - Προσθήκη πηγής *s*
  - Προσθήκη καταβόθρας *t*
  - Μοναδιαία βάρη
- Χρόνος $O((V'+V'')E)$, καθώς
  - το μέγιστο ταίριασμα έχει $(V'+V'')/2$ ακμές, ενώ
  - όλες οι χωρητικότητες ισούνται με 1 $\Rightarrow$ μέγιστη ροή $(V'+V'')/2$



Ροές Ελαχίστου Κόστους

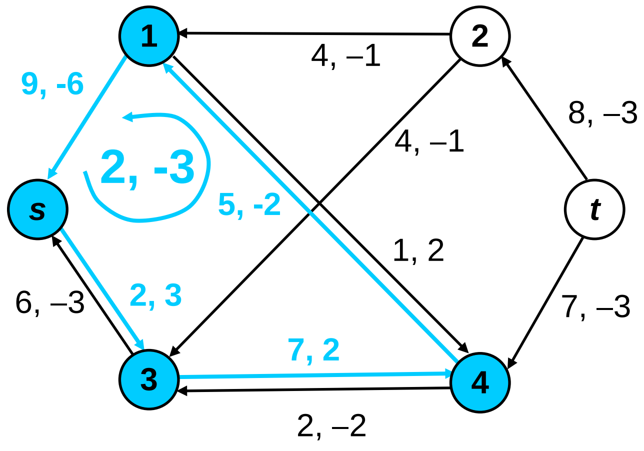
- Ορισμοί
  - Με κάθε ακμή e συσχετίζεται και μία τιμή κόστους $b(e)$ ανά μονάδα ροής
  - Κόστος μίας ροής f:

$$\sum f(e)b(e)$$

  - Μία ροή χαρακτηρίζεται ως ***ελαχίστου κόστους*** εάν επιδεικνύει το ***μικρότερο κόστος μεταξύ όλων των ροών της ίδιας τιμής***
  - Μέγιστη ροή ελαχίστου κόστους το ζητούμενο

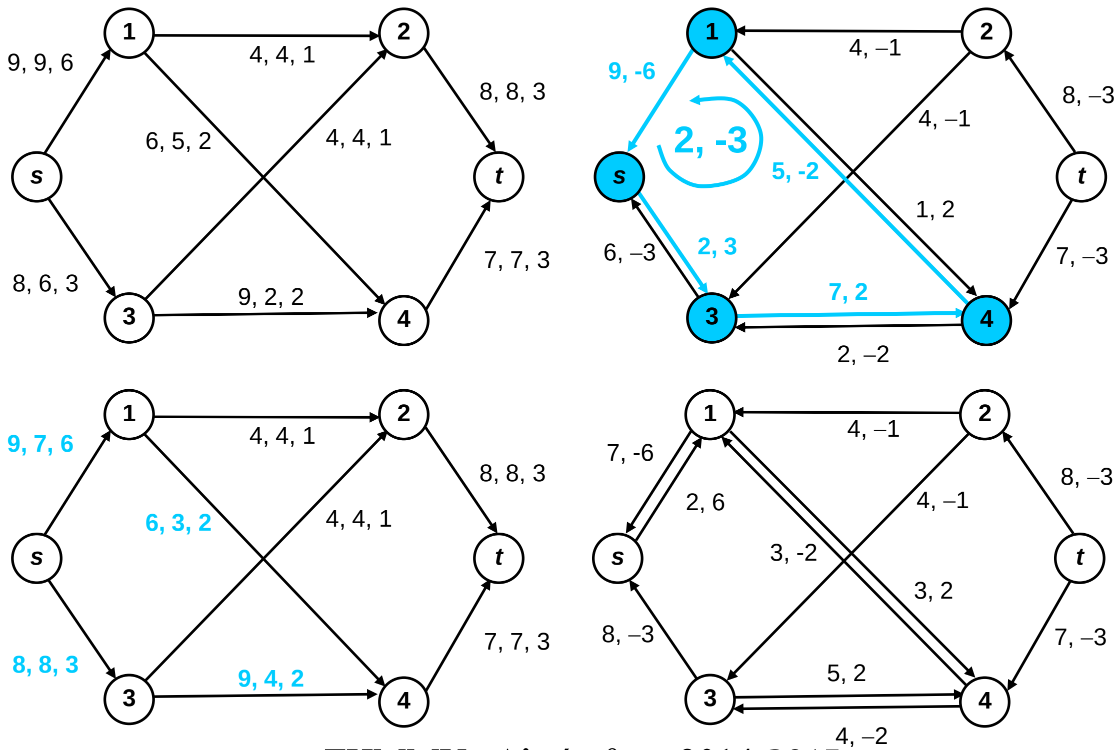
Επέκταση Υπολειμματικού Γραφήματος

- Αναγκαία η επέκταση του υπολειμματικού γραφήματος, ώστε
  – με κάθε πραγματική ακμή *e* να σχετίζεται κόστος *b*(*e*)
  – με κάθε οπισθοακμή, κόστος -*b*(*e*) (καθώς τυχόν επιλογή της ελαττώνει το κόστος)



Πρώτη Λύση

- Χρήση επαυξητικών κύκλων αρνητικού κόστους
  – *Η ύπαρξή τους σημαίνει δυνατότητα μείωσης του κόστους, **χωρίς μεταβολή της τιμής της ροής***

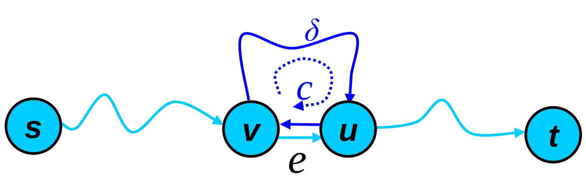


Γιατί δουλεύει;

- Μία ροή είναι ελαχίστου κόστους ανν δεν υπάρχει αρνητικός επαυξητικός κύκλος
  - ($\Rightarrow$) Εάν υπάρχει αρνητικός προφανώς μπορεί κανείς να μειώσει το κόστος
  - ($\Leftarrow$) Εάν η f είναι μέγιστη, όχι ελαχίστου κόστους, αλλά με θετικούς κύκλους, τότε επαύξηση κατά μήκος του κύκλου θα μείωνε το κόστος, δίχως να πειραχθεί η τιμή (άτοπο)
- Πολυπλοκότητα $O(BVE)$, B αρχικό κόστος μέγιστης ροής
  - Edmonds-Karp: για αρχική ροή, κόστους B
  - Bellman-Ford: για εντοπισμό B, το πολύ, αρνητικών κύκλων

Δεύτερη Λύση

- Έστω ροή f ελαχίστου κόστους και επαυξητικό μονοπάτι π ελαχίστου κόστους. Τότε η f': $f \rightarrow_\pi f'$ είναι ροή ελαχίστου κόστους, με μεγαλύτερη, όμως, τιμή ροής:
  - Εάν η f' δεν είναι ελαχίστου κόστους, τότε πρέπει να υπάρχει αρνητικός κύκλος c, με τουλάχιστον μία κοινή ακμή με την π (γιατί; οι δύο ροές διαφέρουν μόνο στις ακμές $\pi = f' - f$ και, άρα, σε αντίθετη περίπτωση ο κύκλος θα υπήρχε και στην f):



  - $b(\underline{s}\rightarrow v)+b(\delta)+b(u\rightarrow t) < b(s\rightarrow v)+b(v\rightarrow u)+b(u\rightarrow t)$, καθώς $b(\delta)+ b(u\rightarrow v) < 0 \Rightarrow b(\delta)-b(v\rightarrow u) < 0 \Rightarrow b(\delta) < b(v\rightarrow u)$

Ανάλυση

- Πολυπλοκότητα
  - $O(|f|VE)$: Bellman-ford για τα ΣΜΜΠ, βάσει της b
  - $O(|f|E\log V)$: Εναλλακτικά, για αραιά γραφήματα, Johnson