Ελάχιστα μονοπάτια για κάθε ζεύγος κόμβων

Είσοδος: Ένα κατευθυνόμενο, με βάρη, γράφημα $G = (V, E)$ με συνάρτηση βάρους $w : E \to \mathcal{R}$.

Έξοδος Για κάθε ζεύγος κόμβων $u, v \in V$ το ελάχιστο μονοπάτι από τον u προς τον v.

Εφαρμογή Να δημιουργηθεί ένας πίνακας αποστάσεων μεταξύ των πόλεων ενός οδικού χάρτη.

Ένας απλός αλγόριθμος

$Trivial\_All\_Pairs(G, w)$

for each $v \in V(G)$ **do**
Run $Dijkstra()$ with v as the source vertex

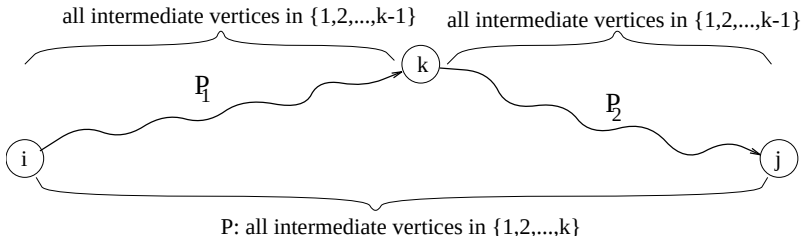
Πολυπλοκότητα Υλοποίηση 'διανύσματος': $O(V^3)$
Υλοποίηση 'δυαδικού σωρού': $O(VE \log V)$

Σημείωση Ο παραπάνω αλγόριθμος δίνει τη σωστή απάντηση μόνο εάν όλα τα βάρη είναι μη-αρνητικά.

Μία προσέγγιση μέσω δυναμικού προγραμματισμού

- **Η δομή ενός ελάχιστου μονοπατιού**

Θεωρήστε το ελάχιστο μονοπάτι από τον κόμβο i προς τον κόμβο j **που διέρχεται μόνο** από κόμβους του συνόλου $\{1, 2, \cdots, k\}$.



- **Μία αναδρομική λύση**

Συμβολισμός $d_{i,j}^{(k)}$ =Το βάρος του ελάχιστου μονοπατιού από τον i προς τον j που διέρχεται μόνο από κόμβους του συνόλου $\{1, 2, \cdots, k\}$.

$$d_{i,j}^{(k)} = \begin{cases} w_{i,j} & \text{εάν } k = 0 \\ \min(d_{i,j}^{(k-1)}, d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)}) & \text{εάν } k \geq 1 \end{cases}$$

$Floyd\_Warshal(W)$

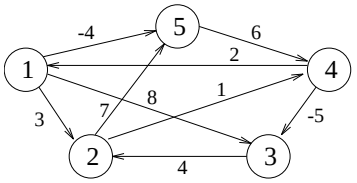
$n = rows(W)$
$d^{(0)} = W$
for $k = 1$ **to** n **do**
  for $i = 1$ **to** n **do**
    for $j = 1$ **to** n **do**
      $d^{(k)}_{i,j} = \min(d^{(k-1)}_{i,j}, d^{(k-1)}_{i,k} + d^{(k-1)}_{k,j})$
return $d^{(n)}$

Πολυπλοκότητα χρόνου & χώρου: $O(V^3)$

Σημείωση 1 Η πολυπλοκότητα χώρου μπορεί να μειωθεί σε $O(V^2)$. Παρατηρήστε ότι ο υπολογισμός του $d^{(k)}$ χρησιμοποιεί μόνο τον $d^{(k-1)}$.

Σημείωση 2 Ο αλγόριθμος $Floyd\_Warshal$ δίνει το σωστό αποτέλεσμα μόνο για γραφήματα **χωρίς αρνητικούς κύκλους**. (Γιατί;)

Σημείωση 3 Έως τώρα, υπολογίσαμε μόνο το κόστος των ελάχιστων μονοπατιών. Δεν έχουμε δείξει πως να ανακτήσουμε τα μονοπάτια. (Με ποιο τρόπο ανακτάμε τα μονοπάτια;)



$$d^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$d^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$d^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$d^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$d^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$d^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

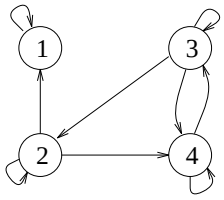
Μεταβατική κλειστότητα κατευθυνόμενων γραφημάτων

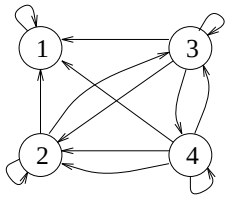
Transitive closure of directed graphs

Είσοδος: Ένα κατευθυνόμενο γράφημα $G = (V, E)$.

Έξοδος: Ένα κατευθυνόμενο γράφημα $G^\star = (V, E^\star)$ όπου
$E^\star = \{(i, j) : \text{υπάρχει μονοπάτι από τον } i \text{ προς τον } j \text{ στο } G\}$.

Παράδειγμα

G: 

G*: 

Ένας απλός αλγόριθμος

- Καταχώρησε βάρος '1' σε όλες τις ακμές του G.
- Εκτέλεσε τον αλγόριθμο $Floyd\_Warshal()$.
- Εάν το κόστος του μονοπατιού από τον i προς τον j είναι διάφορο του ∞ τότε $(i, j) \in E^\star$.

Ένας συναφής αλγόριθμος

$$t_{i,j}^{(0)} = \begin{cases} 0 & \text{εάν } i \neq j \text{ και } (i,j) \notin E \\ 1 & \text{εάν } i = j \text{ ή } (i,j) \in E \end{cases} \qquad t_{i,j}^{(k)} = t_{i,j}^{(k-1)} \ \vee \ (t_{i,k}^{(k-1)} \ \wedge \ t_{k,j}^{(k-1)})$$

$Transitive\_Closure(G)$

$n = |V|$
for $i = 1$ **to** n **do**
  for $j = 1$ **to** n **do**
    if $i = j$ **or** $(i, j) \in E$ **then** $t_{i,j}^{(0)} = 1$
    else $t_{i,j}^{(0)} = 0$
for $k = 1$ **to** n **do**
  for $i = 1$ **to** n **do**
    for $j = 1$ **to** n **do**
      $t_{i,j}^{(k)} = t_{i,j}^{(k-1)} \ \vee \ (t_{i,k}^{(k-1)} \ \wedge \ t_{k,j}^{(k-1)})$
return $t^{(n)}$

Ανάλυση: Πανομοιότυπη με αυτή του $Floyd\_Warshal$.

Πλεονεκτήματα Είναι γρηγορότερος και χρησιμοποιεί λιγότερη μνήμη (κατά ένα σταθερό παράγοντα).