POLITECHNIKA KRAKOWSKA
KATEDRA AUTOMATYKI I TECHNIK INFORMACYJNYCH

Badania Operacyjne

www.pk.edu.pl/~zk/BO_HP.html

Wykładowca:
dr inż. Zbigniew Kokosiński
zk@pk.edu.pl

Wykład 7: Najkrótsze ścieżki w grafie 2.

1. Najkrótsze ścieżki pomiędzy wszystkimi parami wierzchołków.
2. Rozwiązanie rekurencyjne 1 problemu.
3. Algorytm 1.
4. Przykład obliczeniowy.
5. Rozwiązanie rekurencyjne 2 problemu.
6. Algorytm 2 (Floyda-Warshall).
7. Przykład obliczeniowy.
8. Złożoność obliczeniowa algorytmów.

Rozwiązanie rekurencyjne 1

Dany jest graf zorientowany G(V,E), ważony.

Niech teraz $d_{ij}^{(m)}$ będzie najmniejszą wagą ścieżki z wierzchołka i do wierzchołka j spośród tych, które zawierają co najwyżej m krawędzi. Gdy $m = 0$, to istnieje ścieżka z i do j nie zawierająca krawędzi wtedy i tylko wtedy, gdy $i = j$. Zatem

$$d_{ij}^{(0)} = \begin{cases} 0, & \text{jeśli } i = j \\ \infty, & \text{jeśli } i \neq j \end{cases}$$

$$\begin{aligned} d_{ij}^{(m)} &= \min\left(d_{ij}^{(m-1)}, \min_{1 \leqslant k \leqslant n} \{ d_{ik}^{(m-1)} + w_{kj} \} \right) \\ &= \min_{1 \leqslant k \leqslant n} \{ d_{ik}^{(m-1)} + w_{kj} \} \end{aligned}$$

Algorytm 1

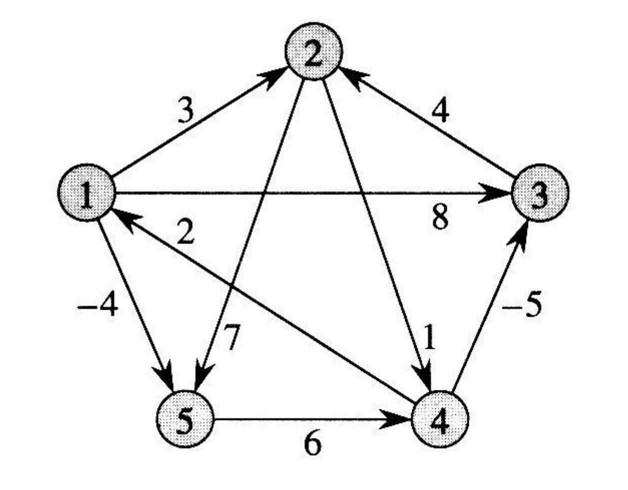
SLOW-ALL-PAIRS-SHORTEST-PATHS(W)
$n \leftarrow rows[W]$
$D^{(1)} \leftarrow W$
for $m \leftarrow 2$ **to** $n - 1$
    do $D^{(m)} \leftarrow$ EXTEND-SHORTEST-PATHS($D^{(m-1)}, W$)
return $D^{(n-1)}$

EXTEND-SHORTEST-PATHS(D, W)
$n \leftarrow rows[\mathrm{D}]$
niech $D' = (d'_{ij})$ będzie macierzą wymiaru $n \times n$
for $i \leftarrow 1$ **to** n
    do for $j \leftarrow 1$ **to** n
        do $d'_{ij} \leftarrow \infty$
            for $k \leftarrow 1$ **to** n
                do $d'_{ij} \leftarrow \min(d'_{ij}, d_{ik} + w_{kj})$
return D'

Algorytm 1 - przykład

EXTEND-SHORTEST-PATHS(D, W)

$n \leftarrow rows[\mathrm{D}]$
niech $D' = (d'_{ij})$ będzie macierzą wymiaru $n \times n$
for $i \leftarrow 1$ **to** n
  do for $j \leftarrow 1$ **to** n
    do $d'_{ij} \leftarrow \infty$
      for $k \leftarrow 1$ **to** n
        do $d'_{ij} \leftarrow \min(d'_{ij}, d_{ik} + w_{kj})$
return D'

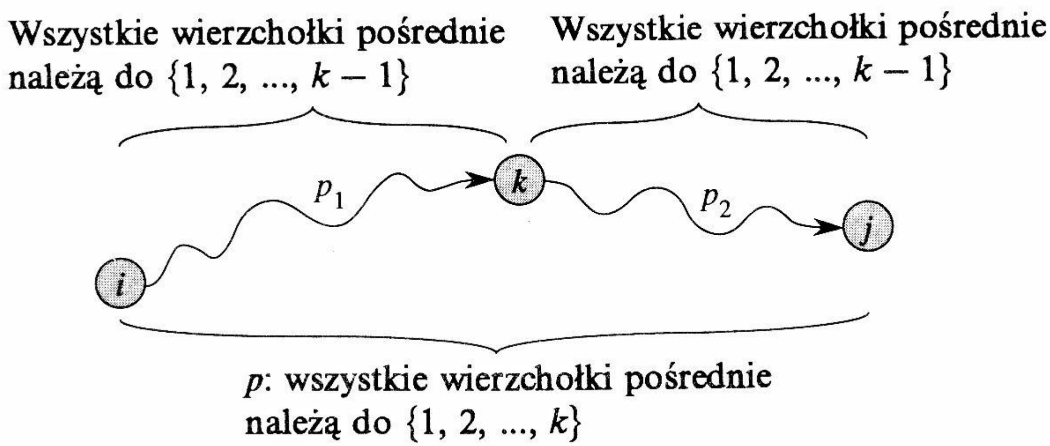


$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11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

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

Idea algorytmu Floyda-Warshalla

Założenie : Graf G(V, E) nie zawiera cykli o ujemnej długości.



Rys. 26.3. Ścieżka p jest najkrótszą ścieżką z wierzchoła i do wierzchołka j, a k jest wewnętrznym wierzchołkiem ścieżki p o największym numerze. Ścieżka p_1, fragment ścieżki p od wierzchołka i do wierzchołka k, ma wszystkie wewnętrzne wierzchołki w zbiorze $\{1, 2, ..., k-1\}$. To samo zachodzi dla ścieżki p_2 prowadzącej z wierzchołka k do wierzchołka j

Rozwiązanie rekurencyjne 2

Macierz $d_{ij}^{(k)}$ definiujemy rekurencyjnie, jak następuje:

$$d_{ij}^{(k)} = \begin{cases} w_{ij}, & \text{jeśli } k = 0 \\ \min(d_{ij}^{k-1},\ d_{ik}^{k-1} + d_{kj}^{k-1}), & \text{jeśli } k \geqslant 1 \end{cases}$$

Macierz $D^{(n)} = (d_{ij}^{(n)})$ jest poszukiwanym rozwiązaniem – $d_{ij}^{(n)} = \delta(i, j)$ dla wszystkich $i, j \in V$ – ponieważ wszystkie wewnętrzne wierzchołki należą do zbioru $\{1, 2, ..., n\}$.

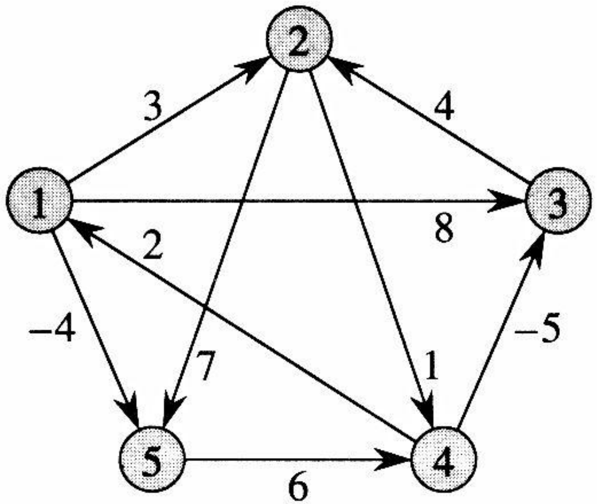
Algorytm Floyda-Warshalla

Założenie : Graf G(V, E) nie zawiera cykli o ujemnej długości.

FLOYD-WARSHALL(W)

$n \leftarrow rows[W]$
$D^{(0)} \leftarrow W$
for $k \leftarrow 1$ **to** n
    do for $i \leftarrow 1$ **to** n
        do for $j \leftarrow 1$ **to** n
            $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1})$
return $D^{(n)}$

Algorytm Floyda-Washalla - przykład



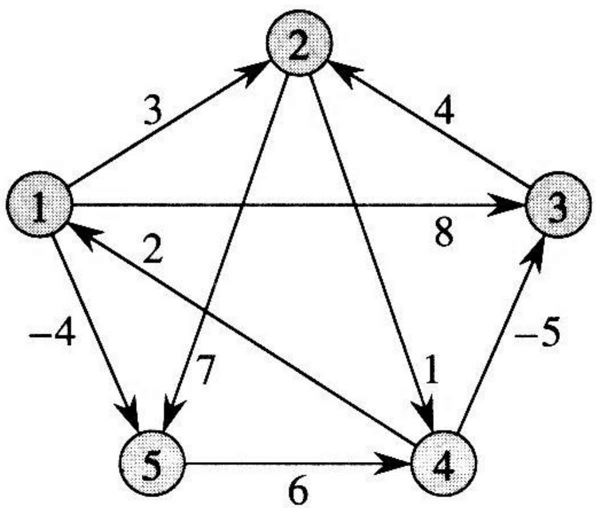
$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \Pi^{(1)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

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

Π – macierz poprzedników, stosowana do wypisania wszystkich wierzchołków żądanej najkrótszej ścieżki

Algorytm Floyda-Washalla – przykład cd.



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

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(4)} = \begin{pmatrix} \mathrm{NIL} & 1 & 4 & 2 & 1 \\ 4 & \mathrm{NIL} & 4 & 2 & 1 \\ 4 & 3 & \mathrm{NIL} & 2 & 1 \\ 4 & 3 & 4 & \mathrm{NIL} & 1 \\ 4 & 3 & 4 & 5 & \mathrm{NIL} \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(5)} = \begin{pmatrix} \mathrm{NIL} & 3 & 4 & 5 & 1 \\ 4 & \mathrm{NIL} & 4 & 2 & 1 \\ 4 & 3 & \mathrm{NIL} & 2 & 1 \\ 4 & 3 & 4 & \mathrm{NIL} & 1 \\ 4 & 3 & 4 & 5 & \mathrm{NIL} \end{pmatrix}$$

Π – macierz poprzedników, stosowana do wypisania wszystkich wierzchołków żądanej najkrótszej ścieżki

Odtwarzanie najkrótszej ścieżki przy pomocy macierzy Π

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
    if i = j
        then wypisz i
        else if π_ij = NIL
            then wypisz "nie ma ścieżki z" i "do" j
            else PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, π_ij)
                wypisz j
```

Złożoność algorytmów 1 i Floyda-Warshalla

Algorytm 1 posiada złożoność $\Theta(n^4)$, ze względu na cztery zagnieżdżone pętle, każda o złożoności $O(n)$.

Algorytm Floyd-Warshalla posiada złożoność $\Theta(n^3)$, ze względu na trzy zagnieżdżone pętle, każda o złożoności $O(n)$.

Źródła wzorów, przykładów i rysunków :

1. Cormen T.H., Leiserson C.E., Rivest R.L. : Wprowadzenie do algorytmów, wyd. 1, WNT, Warszawa, 1997.