O algoritmo de Floyd-Warshall

Dado um grafo direcionado $D(V, E)$ com pesos positivos nas arestas. Dada uma matriz $W = (\omega_{ij})_{n \times n}$ onde
$$\omega_{ij} = \begin{cases} 0, & i = j \\ \omega(i,j), & i \neq j \text{ e } (i,j) \in E \\ \infty, & i \neq j \text{ e } (i,j) \notin E \end{cases}$$

Dado um caminho direcionado $P = v_1, v_2, \ldots, v_k$
o comprimento do caminho
$$\omega(P) = \sum_{i=1}^{k-1} \omega(v_i, v_{i+1})$$
O comprimento do menor caminho de u até v é a distância de u até v:

$$d(u,v) = \begin{cases} \min \{ \omega(P) : P \text{ caminho de } u \text{ a } v \}, \\ \quad \text{se } \exists \text{ caminho de } u \text{ a } v \\ \infty, \text{ c.c.} \end{cases}$$

Princípio de otimalidade: Dado um grafo direcionado $D(V,E)$ com pesos positivos nas arestas. Seja $P = v_1, v_2, \ldots, v_k$ um menor caminho de v_1 a v_k.
Chame de P_{ij} o subcaminho de P desde v_i até v_j.
Então P_{ij} é também menor caminho de v_i a v_j.

O algoritmo de Floyd-Warshall tem como entrada a matriz W de pesos e tem como saída a matriz de distâncias, onde cada posição ij contém $d(v_i, v_j)$.

O algoritmo de Floyd-Warshall considera uma sequência de $n+1$ matrizes $n \times n$

$$W^0 = (w_{ij}^0) = (w(i,j)) = W$$

$W^1 = (w_{ij}^1)$ = comprimento de um menor caminho de i até j sujeito à condição que os vértices intermediários pertencem a $\{v_1\}$.

$W^2 = (w_{ij}^2)$ = comprimento ... a $\{v_1, v_2\}$.

$\vdots$

$W^n = (w_{ij}^n)$ = comprimento ... a $\{v_1, v_2, \dots, v_n\}$.

Portanto a solução será dada pela matriz W^n.

O algoritmo a cada interação obtém melhores cotas superiores para cada entrada.
Ao permitir gradativamente um número maior de vértices intermediários, o algoritmo obtém a solução já que o menor caminho de v_i até v_j é o menor dentre todos os caminhos, permitindo quaisquer vértices intermediários.

algoritmo: Floyd-Warshall

Dados: grafo direcionado, matriz $W = (w_{i,j})$ onde

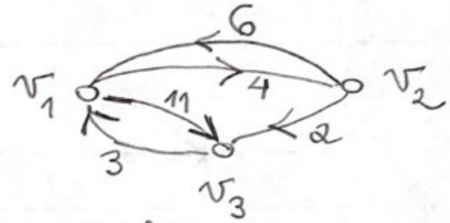
$$w_{ij} = \begin{cases} 0, \text{ se } i = j \\ w(i,j), \text{ se } (i,j) \in E \\ \infty, \text{ c.c} \end{cases}$$

$n :=$ número de colunas de W

$W^0 := W$

para $k = 1$ até n efetuar
    para $i = 1$ até n efetuar
        para $j = 1$ até n efetuar
            $w_{ij}^{k} := \min(w_{ij}^{k-1}, w_{ik}^{k-1} + w_{kj}^{k-1})$

Exemplo



W^0	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

W^1	1	2	3
1	0	4	11
2	6	0	2
3	3	7	0

W^2	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

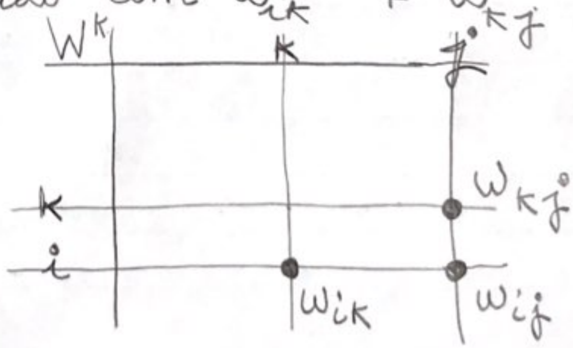
W^3	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

O cálculo requer armazenamento em uma única matriz $n \times n$, onde W^k é obtida de W^{k-1} usando a linha k e a coluna k para revisar os restantes, ie w_{ij}^{k-1} é comparado com $w_{ik}^{k-1} + w_{kj}^{k-1}$

Note que:

$$w_{ik}^{k} = w_{ik}^{k-1}$$

$$w_{kj}^{k} = w_{kj}^{k-1}$$



Teorema: O algoritmo de Floyd-Warshall calcula corretamente a matriz de distâncias

prova: Por indução em k, provaremos que a matriz W^k, $0 \leq k \leq n$, de fato contém na entrada ij o comprimento de um menor caminho de v_i até v_j com vértices intermediários em $\{v_1, v_2, \dots, v_k\}$.

A base é $k=0$ e é válida pela inicialização de W^0; O passo de indução considera a definição da matriz W^k em termos da matriz W^{k-1}, $k>1$.

Temos dois casos:

Caso 1: O menor caminho de v_i até v_j com vértices intermediários em $\{v_1, v_2, \dots, v_k\}$ contém v_k. Pelo princípio de otimalidade, o menor caminho é decomposto em 2 menores caminhos: um de v_i até v_k e outro de v_k até v_j, ambos com vértices intermediários em $\{v_1, v_2, \dots, v_{k-1}\}$, o que por hipótese de indução são as entradas w_{ik}^{k-1} e w_{kj}^{k-1}.

Caso 2: O menor caminho de v_i até v_j com vértices intermediários em $\{v_1, v_2, \dots, v_k\}$ não contém v_k. Logo o menor caminho terá vértices intermediários em $\{v_1, v_2, \dots, v_{k-1}\}$ o que por hipótese de indução é w_{ij}^{k-1}

O Caso 1 corresponde à atualização e o Caso 2 à não atualização de w_{ij}^{k-1}. ☒

A construção dos caminhos mais curtos é feita através de uma sequência de matrizes auxiliares chamadas matrizes predecessores.

$$\pi_{ij}^{0} = \begin{cases} \text{NULO}, & \text{se } i = j \text{ ou } w_{ij} = \infty \\ i, & \text{se } i \neq j \text{ e } w_{ij} < \infty \end{cases}$$

para $K \geq 1$

$$\pi_{ij}^{K} = \begin{cases} \pi_{ij}^{K-1}, & \text{se } w_{ij}^{K-1} \leq w_{iK}^{K-1} + w_{Kj}^{K-1} \\ \pi_{Kj}^{K-1}, & \text{se } w_{ij}^{K-1} > w_{iK}^{K-1} + w_{Kj}^{K-1} \end{cases}$$

para $K = n$ a matriz π_{ij}^{n} contém:
Nulo, se $i = j$ ou se $\nexists$ caminho de i até j
predecessor do vértice j no menor caminho de i até j.

π^0	1	2	3
1	N	1	1
2	2	N	2
3	3	3	N

π^1	1	2	3
1	N	1	1
2	2	N	2
3	3	1	N

π^2	1	2	3
1	N	1	2
2	2	N	2
3	3	1	N

π^3	1	2	3
1	N	1	2
2	3	N	2
3	3	1	N

Programação Dinâmica:
Decomposição do Problema em subproblemas de natureza semelhante ao original

Armazenamos os subproblemas em uma tabela evitando que um subproblema seja computado repetidas vezes.