

Algoritmos em Grafos

Caminhos Mínimos - Algoritmo de Floyd Warshall

Grupo 02:

Bruno Lírio

Thales Veras

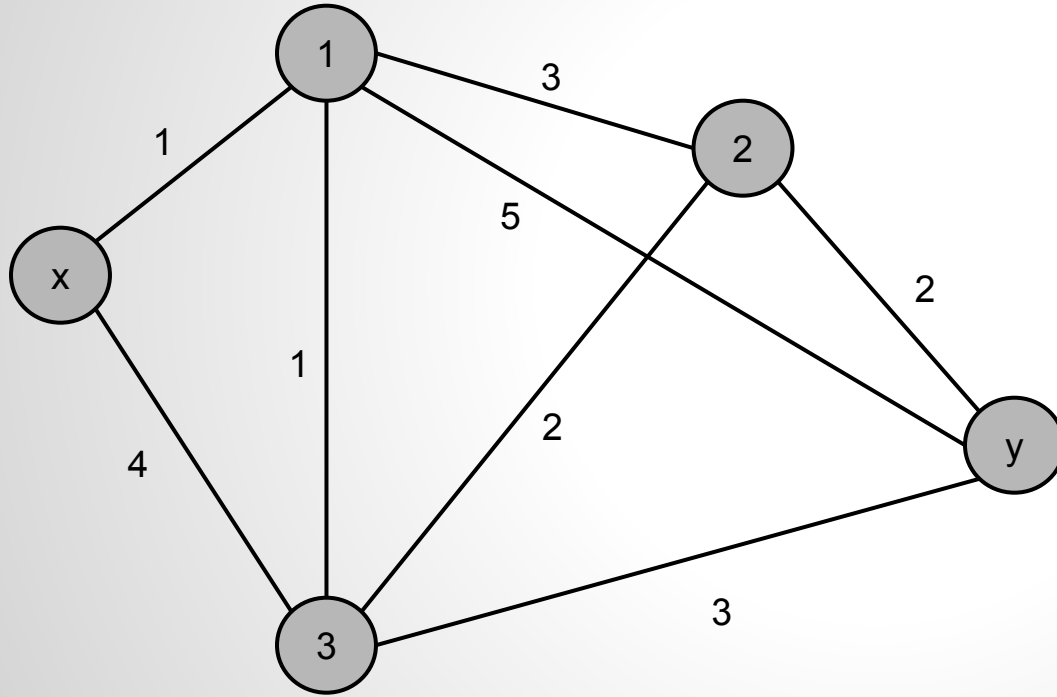
Pedro Paulo

Motivação

O Algoritmo resolve o problema de encontrar o menor caminhos entre todos os pares de nós de um grafo.

As aplicações são muitas mas o problema mais básico no qual podemos pensar é a elaboração de uma tabela com as distâncias entre todos os pares de cidades de um país.

Exemplos Instâncias do Problema



Dado um grafo G direcionado e ponderado, encontrar para todo par u, v de vértices um caminho mínimo de u a v .

Descrição Formal do Algoritmo

FLOYD-WARSHALL(W)

1 $n \leftarrow \text{rows}[W]$

2 $D^{(0)} \leftarrow W$

3 for $k \leftarrow 1$ to n

4 do for $i \leftarrow 1$ to n

5 do for $j \leftarrow 1$ to n

6 do $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

7 return $D^{(n)}$

Descrição da ideia do Algoritmo

A ideia do algoritmo usa a noção de vértices intermediários entre, digamos i e j , que são os vértices para os quais queremos calcular a menor distância, exceto i e j . Então temos um conjunto de vértices intermediários digamos $V = \{1, 2, \dots, k\}$.

Descrição da ideia do Algoritmo

Definida a noção de vértices intermediários, a ideia do algoritmo pode ser descrita da seguinte forma:

Qual é o menor caminho entre i e j sem nenhum vértice intermediário?

Caso em que $k = 0$, e a distância, se houver, é o peso da aresta que liga i a j .

Descrição da ideia do Algoritmo

Qual a menor distância entre i e j , usando 1 vértice intermediário?

Qual a menor distância entre i e j , usando 2 vértices intermediários?

...

Assim quando aumentamos k , temos que pensar apenas sobre o novo vértice.

Descrição da ideia do Algoritmo

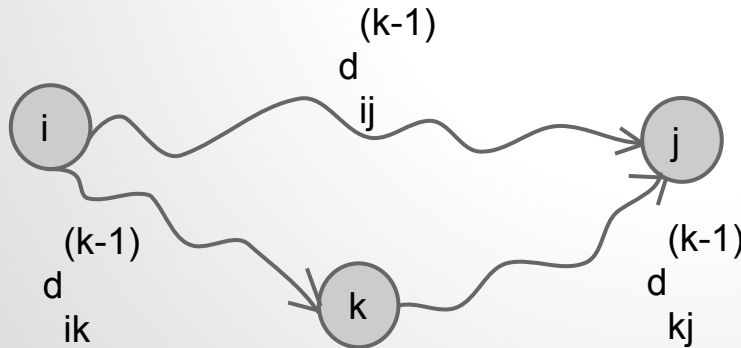
Para resolver o algoritmo a técnica da programação dinâmica é utilizada:

A subestrutura ótima é o fato de que os subcaminhos de caminhos mais curtos são caminhos mais curtos.

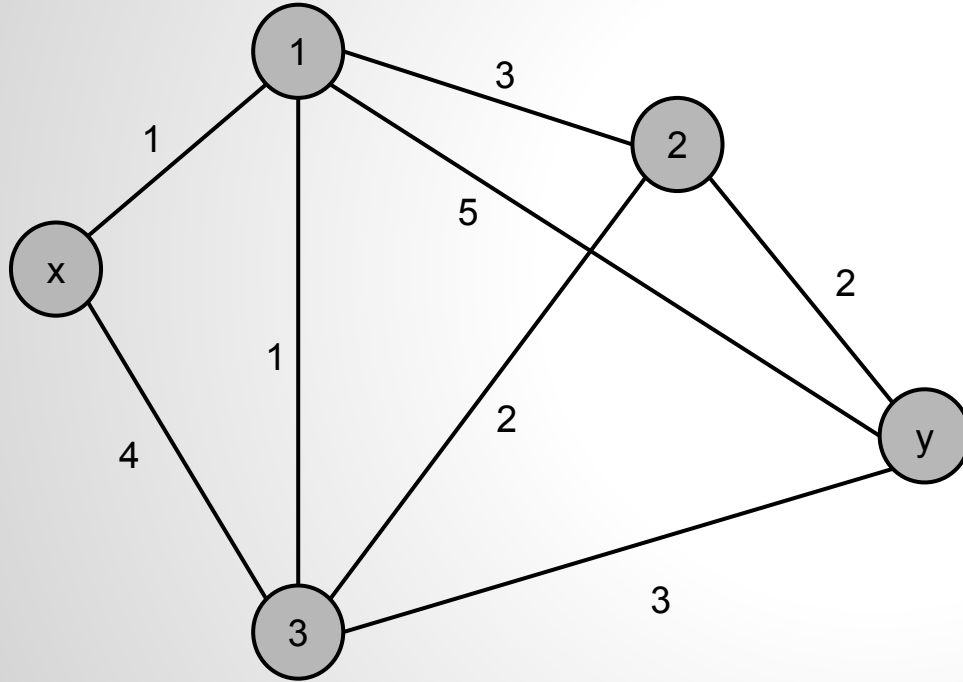
Definição da ideia do Algoritmo

A solução recursiva é:

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0, \\ \min \left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right) & \text{if } k > 0. \end{cases}$$



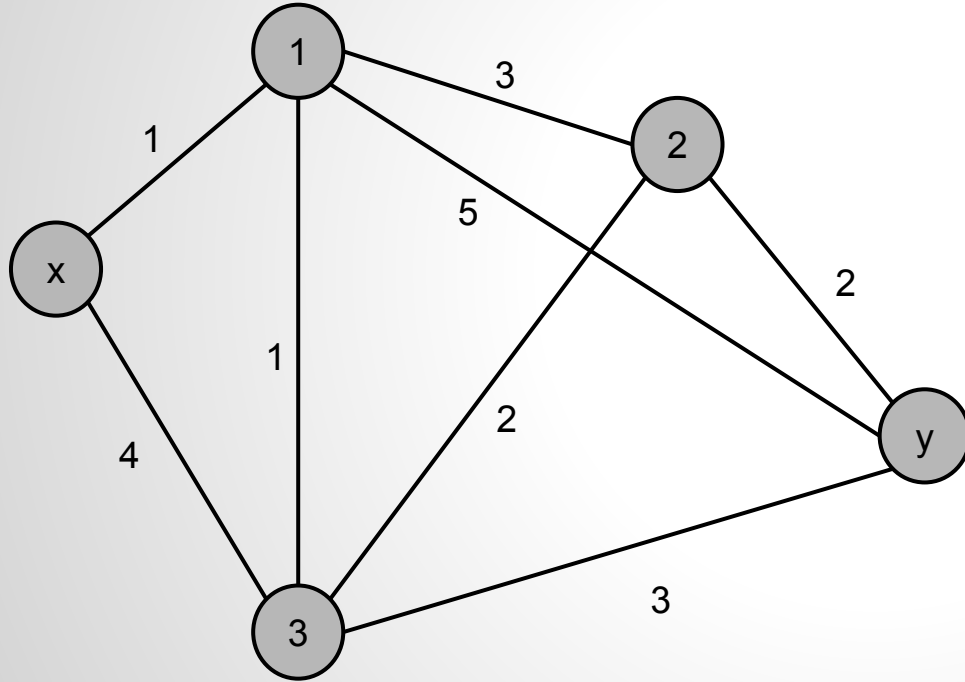
Etapas da Aplicação da Técnica



Inicial: A, após $k = x$

	X	1	2	3	Y
x	0	1	∞	4	∞
1	1	0	3	1	5
2	∞	3	0	2	2
3	4	1	2	0	3
y	∞	5	2	3	0

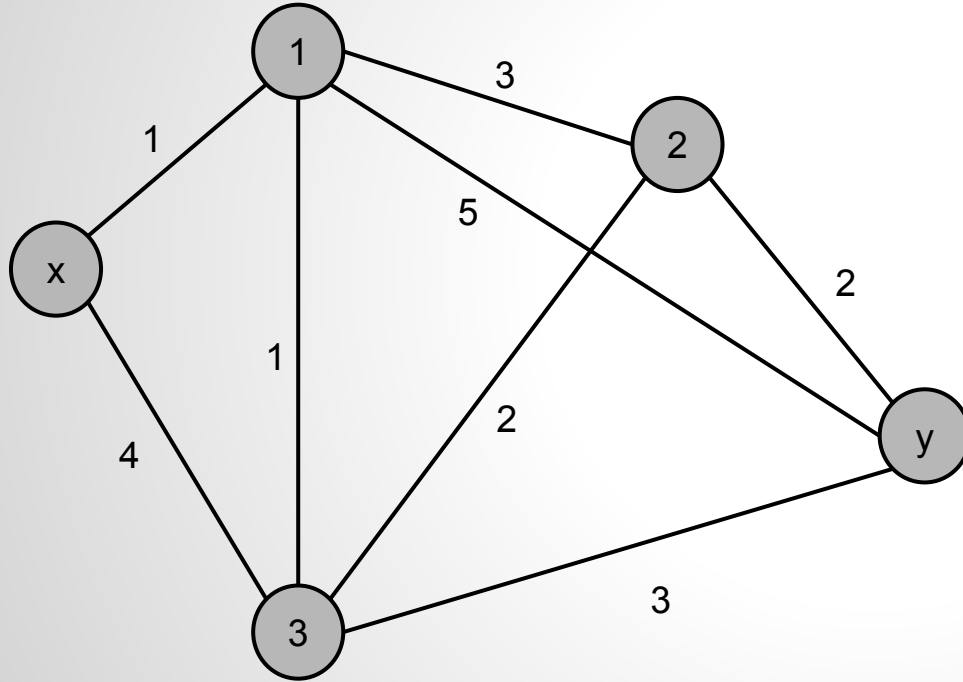
Etapas da Aplicação da Técnica



Após $k = 1$ e $k = 2$

	X	1	2	3	Y
x	0	1	4	2	6
1	1	0	3	1	5
2	4	3	0	2	2
3	2	1	2	0	3
y	6	5	2	3	0

Etapas da Aplicação da Técnica

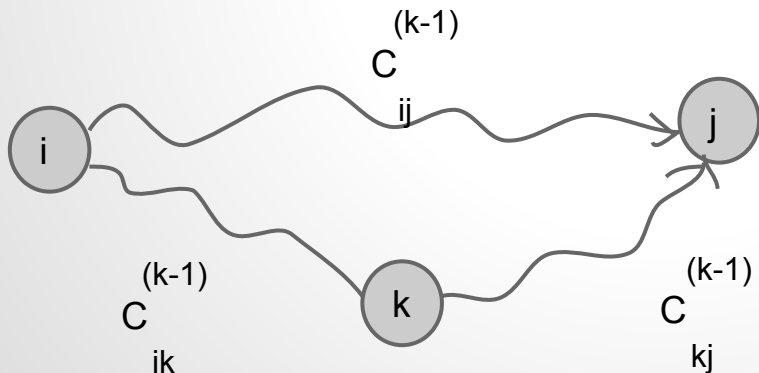


Após $k = 3$ e $k = y$

	X	1	2	3	Y
x	0	1	∞	4	5
1	1	0	3	1	4
2	∞	3	0	2	2
3	4	1	2	0	3
y	5	4	2	3	0

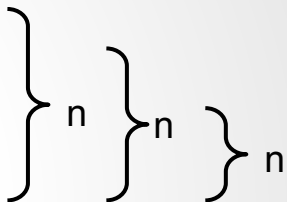
Corretude

A cada iteração o algoritmo compara a menor distância entre os vértices intermediários já calculados e o caminho com a adição do novo vértice intermediário k , selecionando a menor entre elas. Então ao final das iterações, fazendo essas comparações entre todos os vértices, teremos a menor distância.



Complexidade

```
for k:= 1 to n do
  for i:= 1 to n do
    for j:= 1 to n do
      A[i,j] := min (A[i,j], A[i,k] + A[k,j]);
```



Observando o algoritmo acima, é fácil ver, pelos seus 3 loops for, aninhados, que a complexidade é $O(n^3)$.

Conclusão e Discussões

Apesar do algoritmo não fornecer a sequência de arestas dos caminhos mínimos, é muito vantajoso termos a menor distância entre todos os pares de nós de um grafo o que nos possibilita resolver uma série de problemas relacionados a suas aplicações. Um deles, e de grande importância é o fecho transitivo que é uma matriz de 1s e 0s, 1s para os vértices i e j para os quais há um caminho entre eles e 0s para os quais não há. Assim, basta verificar a matriz resultante do algoritmo Floyd-Warshall, se houver algum número para i e j então há um caminho entre eles e a correspondente matriz do fecho transitivo recebe 1 em ij , caso contrário, se for infinito a matriz de fecho transitivo recebe 0.