**SIN110 Algoritmos e Grafos - Exercício E8**

**Aluna:** Caroline Lopes Resek

**Matrícula:** 2017010113

1. Aplique o algoritmo de Dijkstra ao dígrafo abaixo começando com o vértice 1. No começo de cada iteração, dê o custo de cada vértice. No fim da última iteração, exiba a árvore de caminhos mínimos com origem 1.





* Algoritmo de Dijkstra com início no vértice 1:

\* Filas Q: 1 - 5 - 2 - 0 - 4 - 3

5 - 2 - 0 - 4 - 3

2 - 0 - 4 - 3

0 - 4 - 3

4 - 3

3

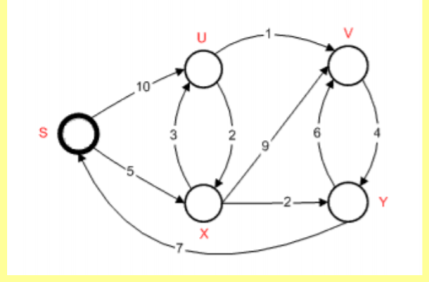
\* Tabela representando a sua execução:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **0** | ~~∞~~ / ~~5~~ / 4 | ~~5~~ / 2 |
| **1** | 0 | - |
| **2** | ~~∞~~ / 3 | 5 |
| **3** | ~~∞~~ / ~~9~~ / 8 | ~~2~~ / 4 |
| **4** | ~~∞~~ / ~~8~~ / 7 | ~~2~~ / 0 |
| **5** | ~~∞~~ / 1 | 1 |

* Árvore de caminhos mínimos com origem 1 :



1. Considere o dígrafo G abaixo:



* 1. Usando o algoritmo de Dijkstra encontre os menores caminhos entre o vértice S e os demais vértices de G.
* Algoritmo de Dijkstra com início no vértice 1:

\* Filas Q: S - U - X - V - Y

X - U - V - Y

Y - U - V

U - V

V

\* Tabela representando a sua execução:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **S** | 0 | - |
| **U** | ~~∞~~ / ~~10~~ / 8 | ~~S~~ / X |
| **X** | ~~∞~~ / 5 | S |
| **V** | ~~∞~~ / ~~14~~ / ~~13~~ / 9 | ~~X~~ / ~~Y~~ / U |
| **Y** | ~~∞~~ / 7 | X |

* Árvore de caminhos mínimos com origem S :



* 1. Modifique o valor (peso) da aresta (X, U) para -3 e empregue o algoritmo de Bellmann-Ford para encontrar os caminhos entre S e os demais vértices. Existem ciclos negativos?



* Algoritmo de Bellmann-Ford com início no vértice S:
* 1ª iteração:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **S** | 0 | - |
| **U** | ~~∞~~ / ~~10~~ / 2 | ~~S~~ / X |
| **X** | ~~∞~~ / 5 | S |
| **V** | ~~∞~~ / 11 | U |
| **Y** | ~~∞~~ / 7 | X |

* 2ª iteração:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **S** | 0 | - |
| **U** | ~~2~~ / 1 | X |
| **X** | ~~5~~ / 4 | ~~S~~ / U |
| **V** | ~~11~~ / 3 | U |
| **Y** | ~~7~~ / 6 | X |

* 3ª iteração:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **S** | 0 | - |
| **U** | ~~1~~ / 0 | X |
| **X** | ~~4~~ / 3 | U |
| **V** | ~~3~~ / 2 | U |
| **Y** | ~~6~~ / 5 | X |

* 4ª iteração:

|  |  |  |
| --- | --- | --- |
| **Vértice** | **d(v)** | **Predecessores** |
| **S** | 0 | - |
| **U** | ~~0~~ / -1 | X |
| **X** | ~~3~~ / 2 | U |
| **V** | ~~2~~ / 1 | U |
| **Y** | ~~5~~ / 4 | X |

* Verifica existência de ciclo negativo :

|  |  |  |  |
| --- | --- | --- | --- |
| **Arco(u-v)** | **Peso** | **d[v] > d[u] + w(u, v)** | **Resposta** |
| **S - U** | 10 | -1 > 10 | Não |
| **S - X** | 5 | 2 > 5 | Não |
| **U -X** | 2 | 2 > 1 | Sim |
| **U - V** | 1 | 1 > 2 | Sim |
| **X - U** | -3 | -1 > -1 | Não |
| **X - V** | 9 | 1 > 11 | Não |
| **X - Y** | 2 | 4 > 4 | Não |
| **Y - S** | 7 | 0 > 11 | Não |
| **Y - V** | 6 | 1 > 10 | Não |
| **V - Y** | 4 | 4 > 5 | Não |

A partir da análise da tabela acima verificou-se que foram encontrados dois ciclos negativos nos arcos (U, X) e (U, V). Vale ressaltar, que o algoritmo ao identificar o primeiro ciclo negativo em (U, X) ele terminaria sua execução, mas continuei a verificação para ver em quais outros arcos seriam identificados outros ciclos negativos caso a execução continuasse. Como foi identificado ciclo negativo, quer dizer que não há solução e, consequentemente, não é possível produzir a árvore de caminhos mínimos

* 1. Comparando as execuções nos itens (a) e (b), a estimativa de tempo de execução é maior em qual algoritmo? Por quê?

O algoritmo (b), pois as suas iterações (execuções) somadas à verificação do ciclo negativo, levam a um consumo maior de tempo do que o algoritmo de Dijkstra.

* 1. Utilize o algoritmo de Floyd-Warshall e encontre o caminho mais curto para todos os pares de vértices de G para o dígrafo acima.



1. Suponha dado um dígrafo com custos não-negativos associados aos vértices (e não aos arcos). O custo de um caminho num tal dígrafo é a soma dos custos dos vértices do caminho. Quero encontrar um caminho de custo mínimo dentre os que começam num vértice s e terminam num vértice t. Adapte o algoritmo de Dijkstra para resolver esse problema.

v = vértice G = grafo chave[u] = custo de u

OrigemÚnica(G,s)

para cada v em G faça

d[v] ∞

pred[v] NIL

d[s] 0

Relaxamento(u,v)

se d[v] > d[u] + chave[v]

então d[v] d[u] + chave[v]

pred[v] u

Arborescência(G,x,v)

se v = x

então imprime x

senão se pred[v] = NIL

então escreve “Não há caminho”

senão Arborescência(G,x, pred(v))

escreve “v ”

Dijkstra(G,s,t)

OrigemÚnica(G,s)

Q CriaFila( )

enquanto Q faça

u RetireMínimo(Q)

para cada v em Adj[u] faça

Relaxamento(u,v)

Arborescência(G,s,v)

