



IPS Instituto
Politécnico de Setúbal
**Escola Superior de
Tecnologia de Setúbal**

Programação Avançada 2021-22

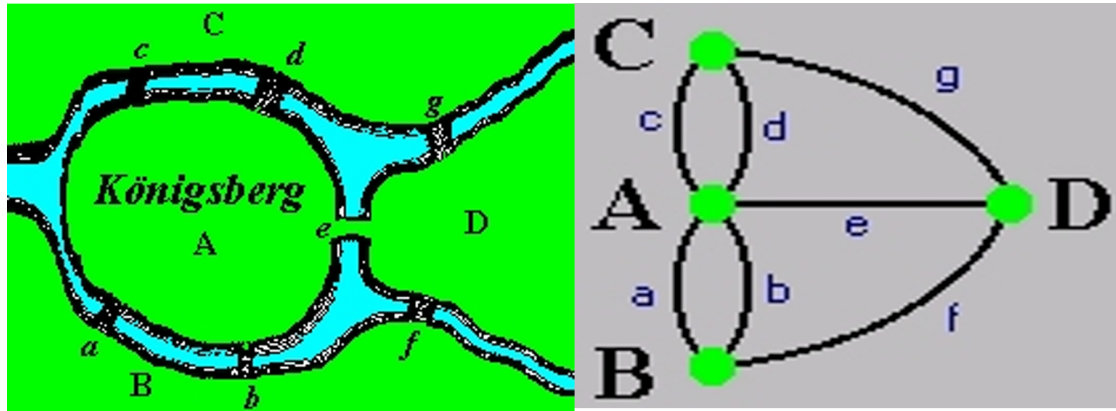
[3c] Grafos | Algoritmo Dijkstra

Bruno Silva, Patrícia Macedo

Sumário

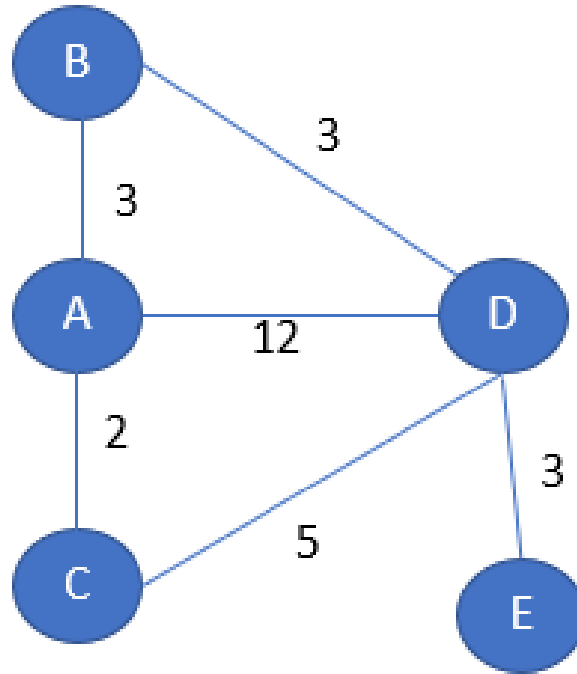
- Noção de **Caminho mais curto** ou de **Menor Custo**;
- Algoritmo *Dijkstra*;
- Algoritmo do Caminho de menor custo;
- Exercícios.

O caminho de menor custo : Conceito



1. Qual o caminho de menor custo para ir de A a D ?
 - Se as pontes não tiverem uma distância ou custo associados, diremos que o caminho de menor custo, é ir de A a D pela ponte e .
2. Se as pontes tiverem portagem (custo associado):
 - (a - 2) (b - 1) (c - 3) (d - 1) (g - 4) (f - 4) (e - 15)
 - Neste caso o caminho de menor custo consiste em seguir pelas pontes d e g para chegar de A a D .

Procura do Caminho de menor custo



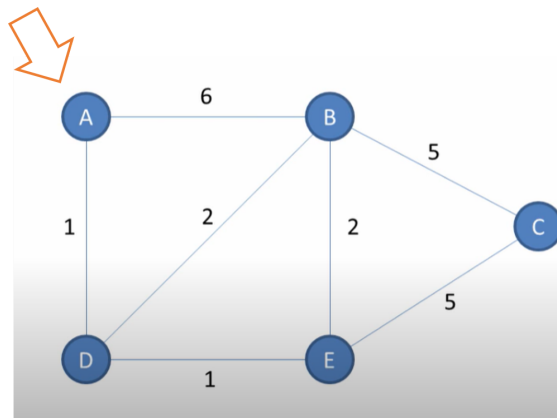
- Qual o caminho de **menor custo** entre A e E?
- Qual o caminho de **menor curso** (que percorre um menor número de arestas) entre A e E ?

Procura do Caminho de menor custo

1. Qual o caminho de menor custo (ou de menor valor) entre dois vértices de um Graph?
 - O caminho de menor custo entre dois pontos, é aquele em que a soma do valor das arestas percorridas é menor.
2. Como se determina o valor de uma aresta?
 - Caso estejamos perante um Graph valorado usa-se o valor associado a cada aresta para determinar o caminho de menor valor entre dois pontos;
 - Nos restantes casos, considera-se que cada aresta percorrida tem o valor 1. O caminho de menor custo é aquele que percorrer menos arestas.

Algoritmo Dijkstra | Conceitos

- O algoritmo de *Dijkstra*, foi publicado em 1959 pelo cientista da computação holandês *Edsger Dijkstra*, soluciona o problema do caminho de menor custo num grafo com arestas de valor não negativo.
- Dado um grafo e um vértice de origem, o algoritmo calcula o menor caminho entre um vértice origem e **todos** os outros vértices do grafo (é um algoritmo apelidado de **guloso**, pois consome muitos recursos computacionais)



Vertex	Shortest distance from A	Previous vertex
A	0	
B	3	D
C	7	E
D	1	A
E	2	D

Algoritmo Dijkstra | Conceitos

Input:

- Um graph valorado $G = (V, E)$:
 - $w : E \rightarrow \mathbb{R}$ (cada aresta tem um valor associado / peso)-
- um nó de partida s .

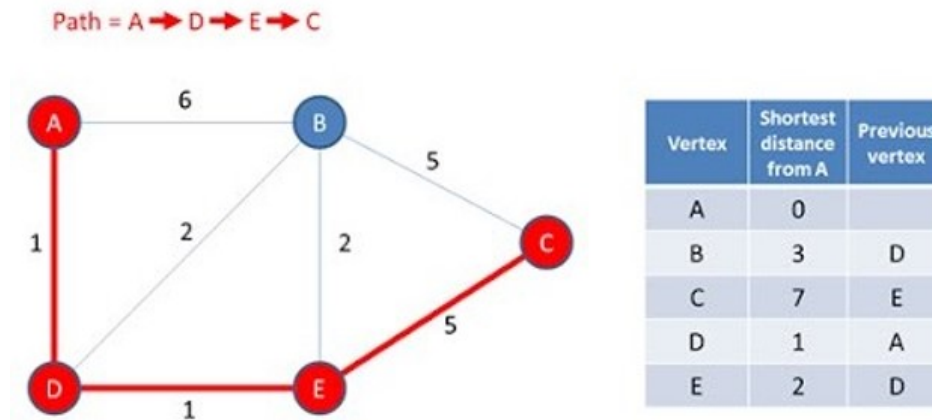
Output:

- Caminho de menor custo de s para **todos os outros nós do graph**.

O custo (ou distância, ou peso) de um caminho é igual ao **somatório** do custo das arestas que constituem o caminho, ou infinito se não existir caminho.

Algoritmo Dijkstra | Em funcionamento

Veja : <https://www.youtube.com/watch?v=pVfj6mxhdMw>



Algoritmo Determinar o caminho de menor custo | Pseudocódigo

1. O algoritmo do cálculo do caminho de menor custo entre dois pontos de um grafo, usa o algoritmo Dijkstra.
2. Assim a execução do algoritmo de Dijkstra é um passo do algoritmo para cálculo de caminho de menor custo e do seu valor.

Algoritmo do caminho de menor custo |Pseudocódigo.

Algorithm: Minumum_Cost_Path

Input - graph, vOri- start vertex ,vDst - destination vertex

Output - paths[] - array_with the sequence of_vertex from vOrig _to vDst
cost - cost associate _to_ the path found

BEGIN

Disjktra(graph,vOri, costs[], predecessors[]) //algorithm

path<-[]

v <- vDst

WHILE v ≠ vOri **DO**

add(path, 0,v)

v<-predecessors[v]

END WHILE

add(path, 0,v)

cost<-costs[vDst]

END Minumum_Cost_Path

Algoritmo Dijkstra | Pseudocódigo

```

Algorithm: Dijkstra
Input - (graph,origin)
Output - costs[] e predecessors[]
BEGIN
    FOR EACH vertex v in graph
        costs[v] <- Infinit
        predecessor[v] <- null
    END FOR
    costs[origin] <- 0
    s <- {all vértices of graph}
    WHILE (s IS NOT EMPTY)
        u <- findLowerVertex(graph, costs[],s) //vertice não visitado de menor custo
        IF (costs[u] = Infinit) THEN RETURN
        removeLowVertex(s,u) // remove vertex u from set s
        FOR EACH v adjacent_of_vertex(u)
            cost <- costs[u] + cost_between(u,v)
            IF (cost < costs[v]) THEN
                costs[v] <- cost;
                predecessor[v] <- u;
            END IF
        END FOR
    END WHILE
END Dijkstra
    
```

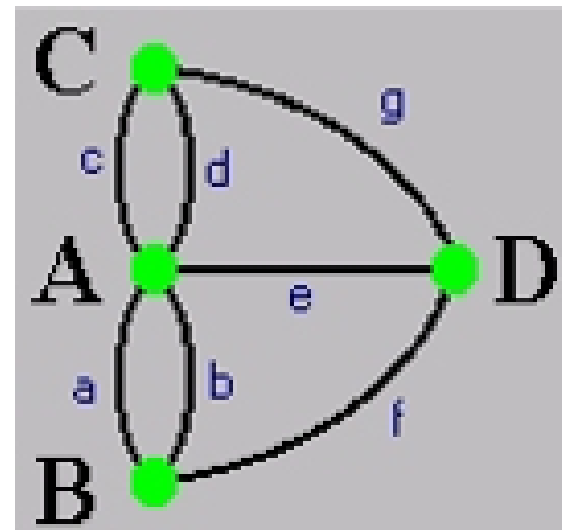
Exercício 2.1

Pretende-se utilizar o algoritmo acima apresentado para calcular o caminho de menor custo, entre dois pontos do grafo abaixo, assumindo os seguintes custos para percorrer as arestas

- (a - 2) (b - 1) (c - 3) (d - 1) (g - 4) (f - 4) (e - 15)

1. Preencha a tabela que mostra o resultado da aplicação do algoritmo de Dijkstra, a partir do Local A.

Vertex	Vertex predecessor	Cost
A		
B		
C		
D		



Exercícios 2.1 (cont)

2. Consultando a tabela construída anteriormente determine

- O Caminho de menor custo, para ir de A a D ? Qual o custo?

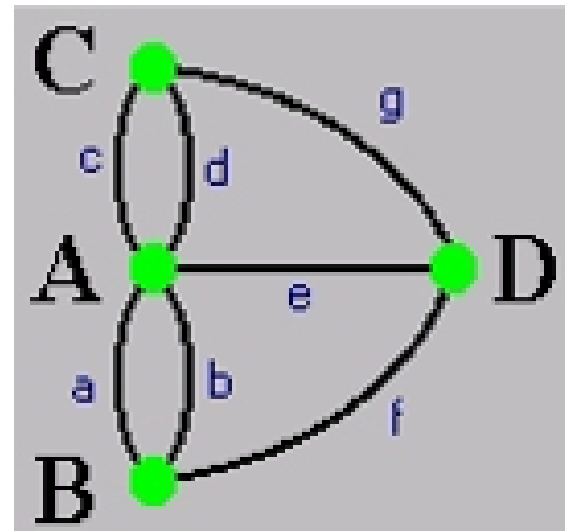
Nota Como saber qual a ponte atravessada se a **a** ou **b** ??

Exercícios 2.1 (cont)

3. De forma a saber qual a ponte percorrida entre dois locais (já que existem arestas paralelas), pretende-se modificar o algoritmo acima apresentado de forma a que este regista não só a lista de locais percorridos, **mas também as arestas percorridas**.

Preencha esta nova a tabela que mostra o resultado da aplicação do algoritmo de Dijkstra , a partir do **Local A** . Note que ela contém mais uma coluna.

<u>Vertex</u>	<u>Vertex predecessor</u>	<u>Edge</u>	<u>Cost</u>
A			
B			
C			
D			



Exercícios 2.1 (cont)

4. Consultando a tabela construindo anteriormente determine
 - A sequência de pontes percorridas para para ir de A a D ao **menor custo**.
 - A sequência de pontes percorridas para para ir de B a C ao **menor custo**.

