ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

P.PORTO

Matemática Discreta 2022/2023

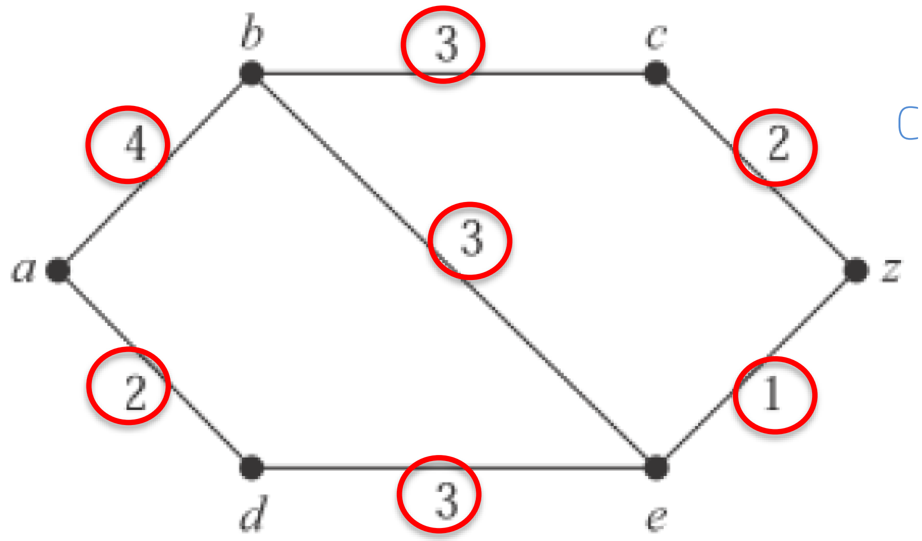
Teoria de Grafos

Caminho mais curto, Árvores

Teste os seus conhecimentos

Faça o Diagnóstico no moodle

Grafo Ponderado



Conjunto dos vértices:

$$V = \{a, b, c, d, e, z\}$$

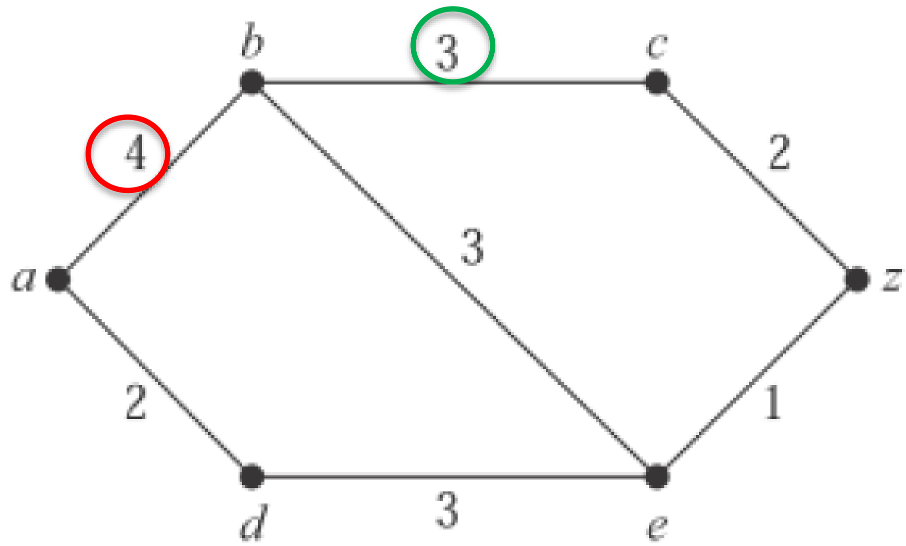
Conjunto das arestas: $E = \{(a,b),(a,d),(b,c),(b,e),(d,e),(c,z),(e,z)\}$

Definição 15:

Consideremos que G é um **grafo ponderado**, i.e. um grafo que a cada aresta, e, é associado valor não negativo, $w(e)$, a que dá-nos o nome de peso ou custo. Podemos representar os pesos de cada arestas recorrendo à chamada **matriz de pesos** (ou matriz de custos), $W = (w_{ij})$, definida por:

$$w_{ij} = \begin{cases} w(e), & \text{se é a aresta do vértice } v_i \text{ para o vértice } v_j \\ 0, & \text{caso não exista uma aresta do vértice } v_i \text{ para o vértice } v_j \end{cases}$$

O problema do caminho mais curto



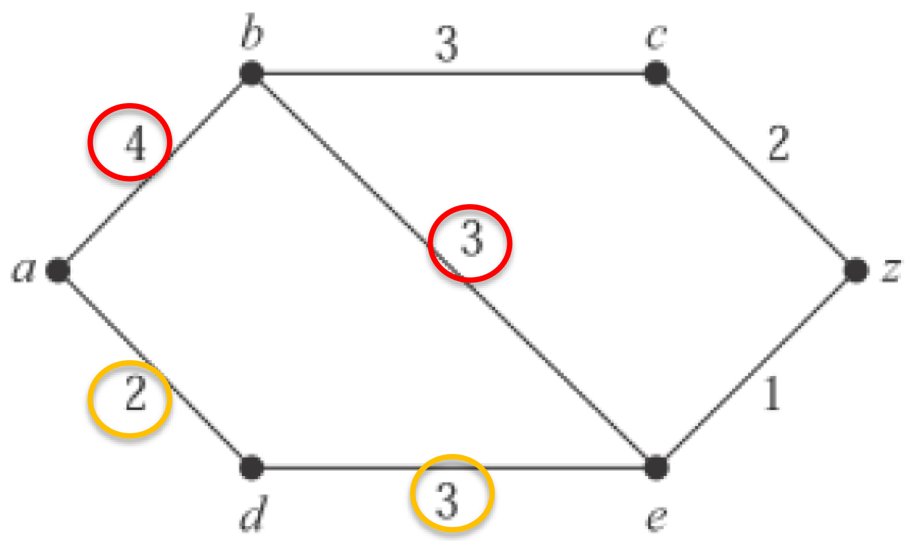
Grafo Ponderado

Matriz dos pesos:

$$W = \begin{bmatrix} 0 & 4 & 0 & 2 & 0 & 0 \\ 4 & 0 & 3 & 0 & 3 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 3 & 0 \\ 0 & 3 & 0 & 3 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

Não existe a aresta (d, b)

O problema do caminho mais curto



Definição 16:

O peso ou custo de um caminho é a soma dos pesos de todas as arestas que o formam.

Por exemplo,

o peso ou custo do caminho a, b, e é

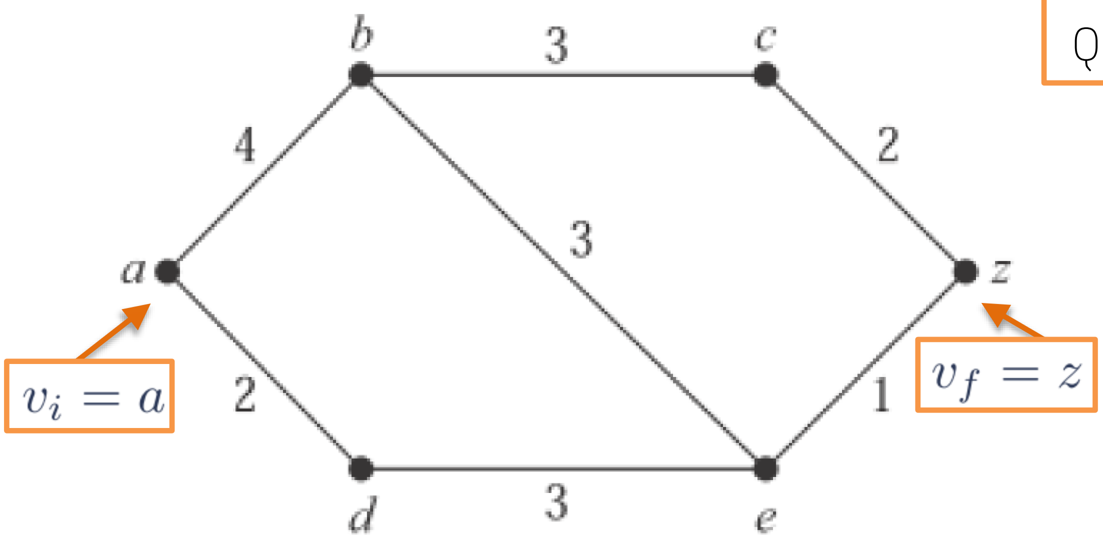
o peso ou custo do caminho a, d, e é

Caminho mais curto de $\boldsymbol{d}$ a $\boldsymbol{e}$

Qual é o caminho mais curto entre as cidades de Bragança e Faro?

O **Algoritmo de Dijkstra** permite resolver este problema do caminho mais curto, para grafos não orientados com **pesos positivos.**

Para grafos com **pesos negativos** pode-se utilizar o algoritmo de **Bellman-Ford**.

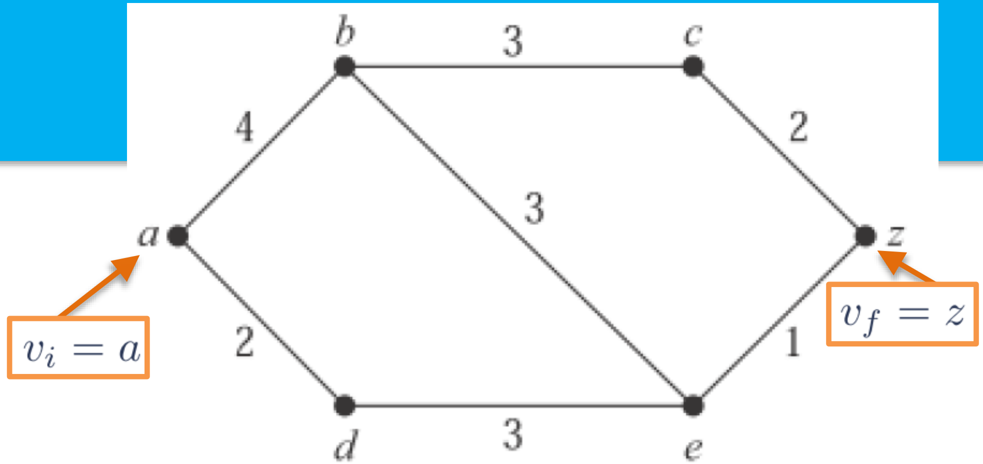


Qual é o caminho mais curto entre $\boldsymbol{a}$ e $\boldsymbol{z}$?

Algoritmo de Dijkstra

Input:

- Grafo simples, não orientado e ponderado G=(V,E)
- Matriz dos pesos
- Vértice inicial $\boldsymbol{v_i}$
- Vértice final $\boldsymbol{v_f}$



Algoritmo de Dijkstra

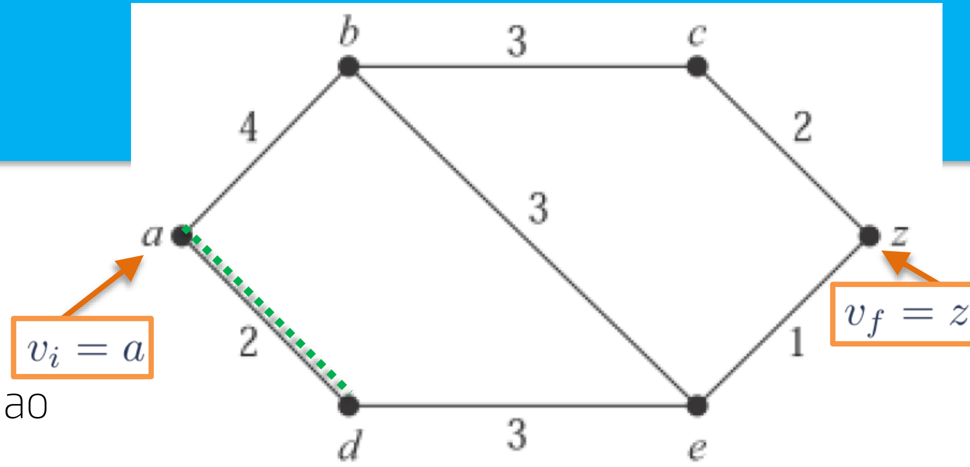
1. Inicializar os conjuntos:
   - M = ∅ conjunto dos vértices já explorados
   - X conjunto dos vértices adjacentes ao vértice inicial v_i
   - Xd conjunto dos pesos dos caminhos de v_i a cada elemento de X
   - R conjunto com todos os caminhos que ligam v_i aos vértices de X

It.	v_d (M)	Mc	A	$v_i, \ldots, v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d

v_d vértice que está a ser explorado

Mc Caminho mínimo de v_i a v_d

A conjunto dos vértices adjacentes a v_d , não pertencentes a M



Algoritmo de Dijkstra

2. **Selecionar** o vértice v_d correspondente ao elemento mínimo de X;

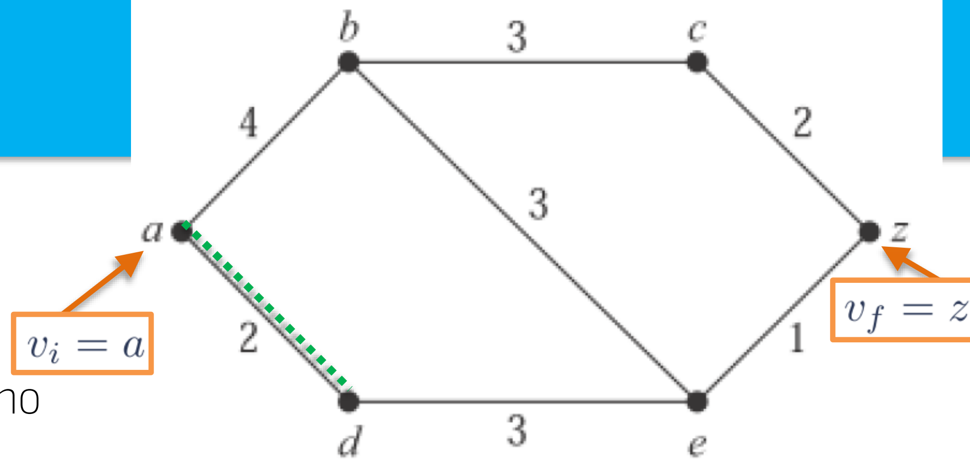
Transferir o vértice v_d para o conjunto M;

Transferir o caminho mais curto entre v_i e v_d para o conjunto Mc.

It.	v_d (M)	Mc	A	$v_i, \ldots, v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d
1	d	a, d	$\{e\}$	$a, d, e \longmapsto L(e) = 2 + 3 = 5$	$\{b, e\}$ $\{4, 5\}$	a, b a, d, e

v_d vértice que está a ser explorado

Mc Caminho mínimo de v_i a v_d



Algoritmo de Dijkstra

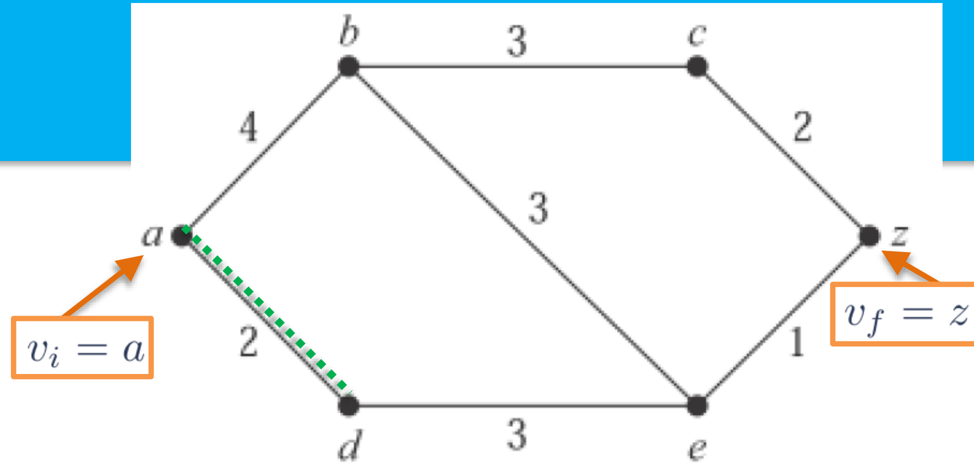
3. Se $v_d = v_f$ então TERMINAR o algoritmo

4. **Selecionar** os vértices adjacentes de v_d não pertencentes a M.

4.1. Determinar as distâncias de v_i até cada um dos vértices do conjunto A, somando à distância de v_i a v_d as distâncias de v_d a esses vértices.

It.	v_d (M)	Mc	A	$v_i, \ldots, v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d
1	d	a, d	$\{e\}$	$a, d, e \longmapsto L(e) = 2 + 3 = 5$	$\{b, e\}$ $\{4, 5\}$	a, b a, d, e

A conjunto dos vértices adjacentes a v_d , não pertencentes a M

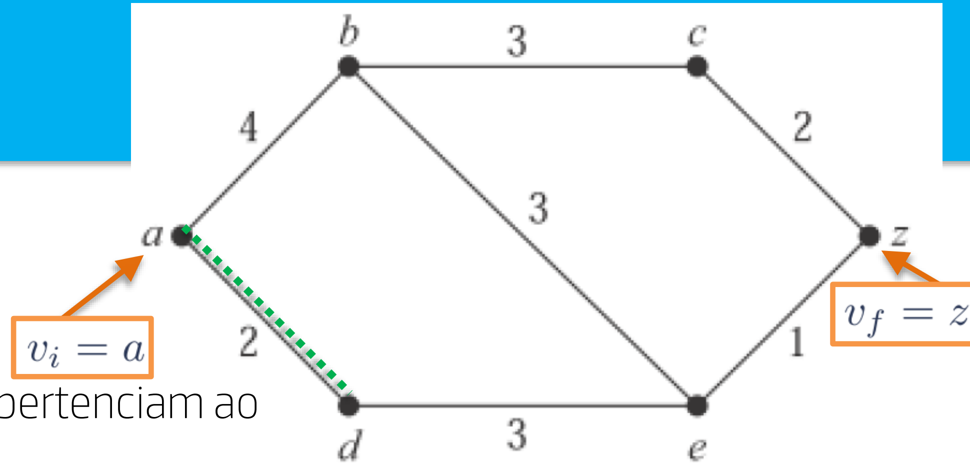


Algoritmo de Dijkstra

4.2. Introduzir

- no conjunto X os vértices do conjunto A que não pertençam a X
- no conjunto Xd as distâncias de $\boldsymbol{v_i}$ a esses vértices;
- no conjunto R os respetivos caminhos.

It.	v_d (M)	Mc	A	$v_i, ..., v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d
1	d	a, d	$\{e\}$	$a, d, e \longmapsto L(e) = 2 + 3 = 5$	$\boxed{\{b, e\}}$ $\boxed{\{4, 5\}}$	a, b a, d, e



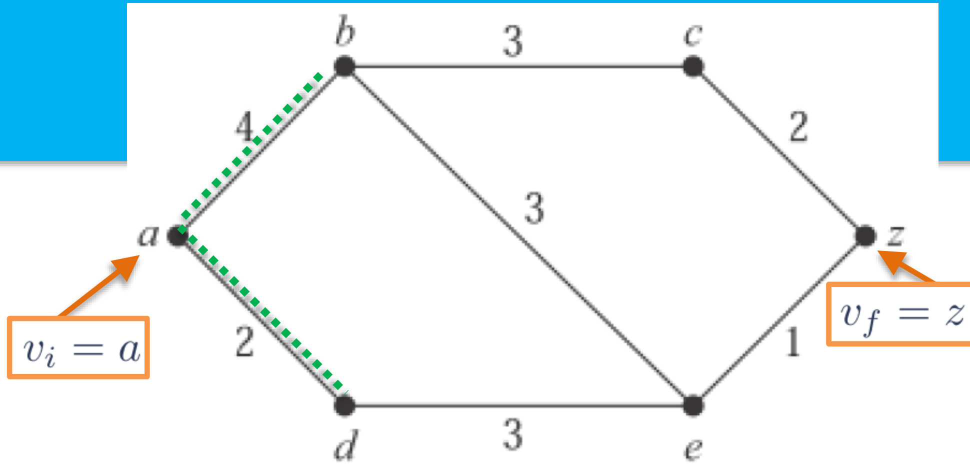
Algoritmo de Dijkstra

4.3. Se houver, no conjunto A, vértices que já pertenciam ao conjunto X, mas com distâncias em Xd superiores às agora determinadas, então, para cada um desses vértices, v_e, **substituir**:

- no conjunto Xd a distância associada ao vértice v_e pela nova distância;
- no conjunto R, o caminho de v_i para v_e, pelo novo caminho v_i, ..., v_d, v_e.

Note que v_i, ..., v_d representa o caminho mais curto de v_i para v_d.

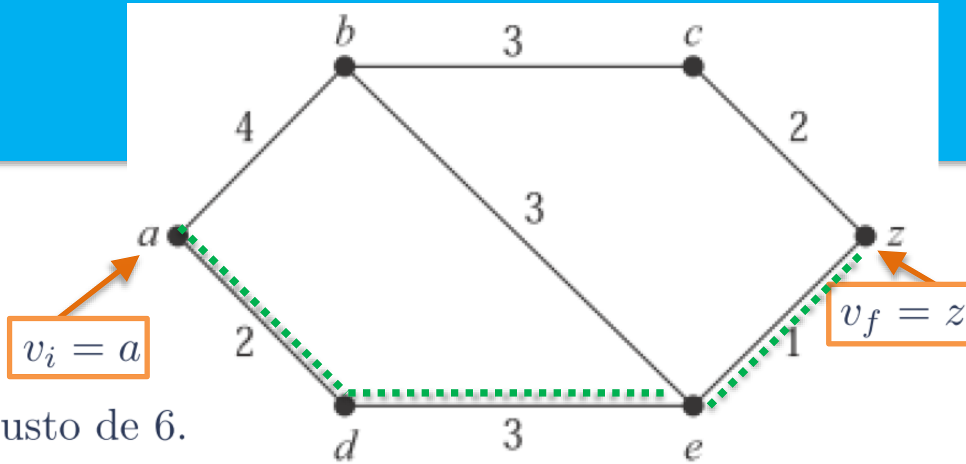
5. Se no conjunto X estiver apenas o vértice v_f , então TERMINAR, senão, voltar a 2.



Algoritmo de Dijkstra

It.	v_d (M)	Mc	A	$v_i, ..., v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d
1	d	a, d	$\{e\}$	$a, d, e \longmapsto L(e) = 2 + 3 = 5$	$\{b, e\}$ $\{4, 5\}$	a, b a, d, e
2	b	a, b	$\{c, e\}$	$a, b, c \longmapsto L(c) = 4 + 3 = 7$ $a, b, e \longmapsto L(e) = 4 + 3 = 7$	$\{e, c\}$ $\{5, 7\}$	a, d, e a, b, c
3	e	a, d, e	$\{z\}$	$a, d, e, z \longmapsto L(z) = 5 + 1 = 6$	$\{z, c\}$ $\{6, 7\}$	a, b, c a, d, e, z

Algoritmo de Dijkstra

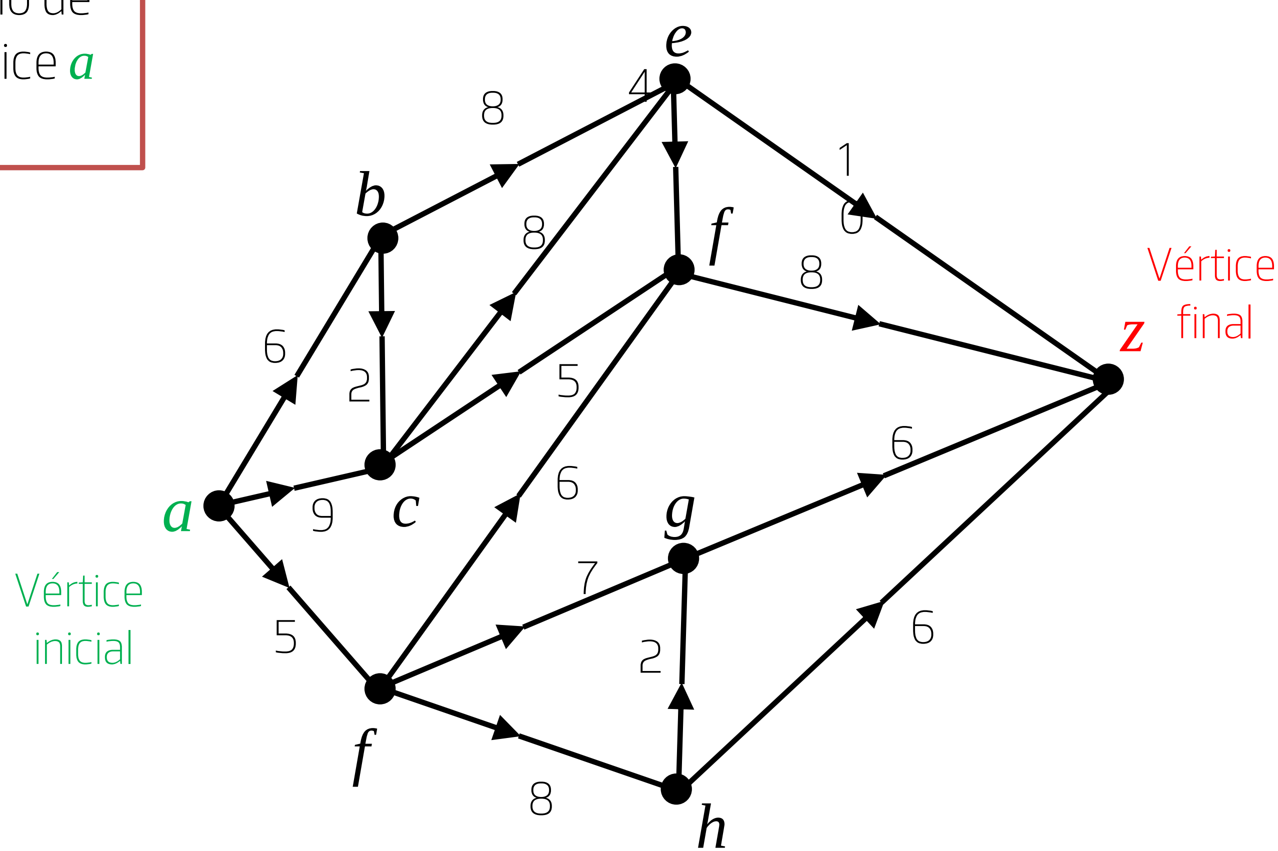


O caminho mínimo é a, d, e, z e tem um custo de 6.

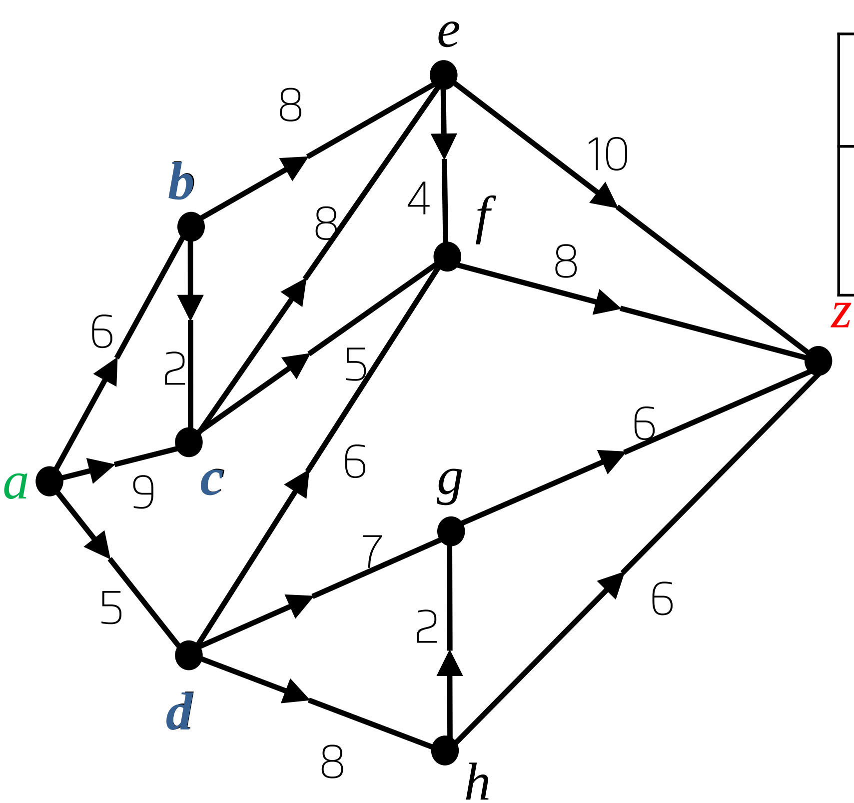
It.	v_d (M)	Mc	A	$v_i, ..., v_d, v_j$ e $L(v_j)$	X e Xd	R
0		a	$\{b, d\}$	$a, b \longmapsto L(b) = 4$ $a, d \longmapsto L(d) = 2$	$\{d, b\}$ $\{2, 4\}$	a, b a, d
1	d	a, d	$\{e\}$	$a, d, e \longmapsto L(e) = 2 + 3 = 5$	$\{b, e\}$ $\{4, 5\}$	a, b a, d, e
2	b	a, b	$\{c, e\}$	$a, b, c \longmapsto L(c) = 4 + 3 = 7$ $a, b, e \longmapsto L(e) = 4 + 3 = 7$	$\{e, c\}$ $\{5, 7\}$	a, d, e a, b, c
3	e	a, d, e	$\{z\}$	$a, d, e, z \longmapsto L(z) = 5 + 1 = 6$	$\{z, c\}$ $\{6, 7\}$	a, b, c a, d, e, z

Algoritmo de Dijkstra – Digrafo

Determine o caminho de menor custo do vértice a ao vértice z

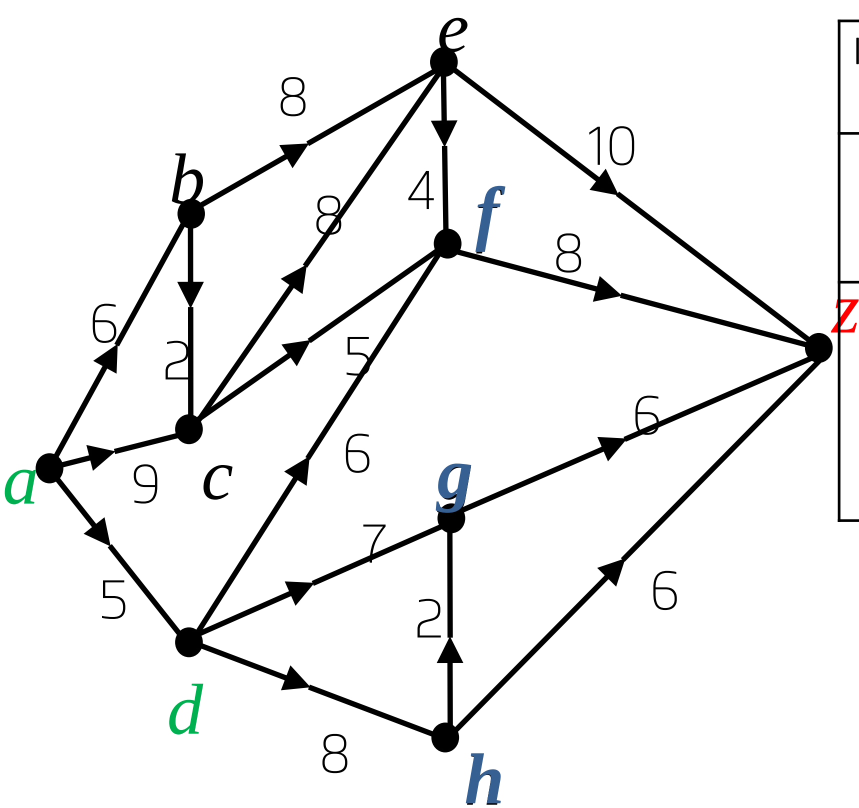


Algoritmo de Dijkstra – Problema do caminho mais curto



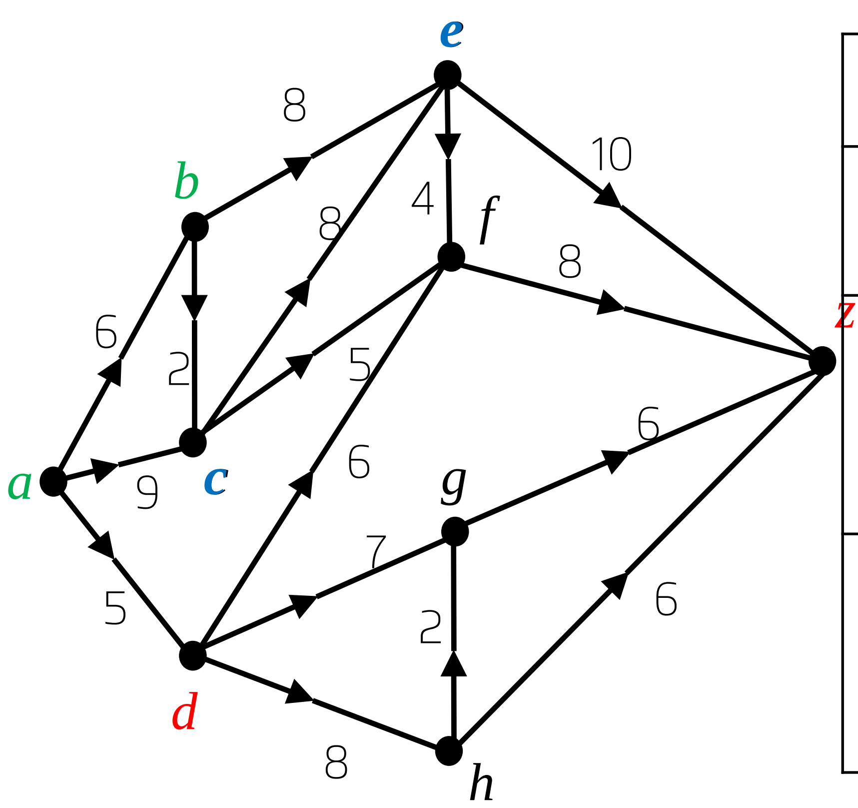
It.	v_d (M)	Mc	A	$v_i, \dots, v_d, v_j$ L(v_j)	X Xd	R
0	---	a	$\{b,c,d\}$	$a,b \rightarrow 6$ $a,c \rightarrow 9$ $a,d \rightarrow 5$	$\{d,b,c\}$ $\{5,6,9\}$	a,b a,c a,d

Algoritmo de Dijkstra – Problema do caminho mais curto



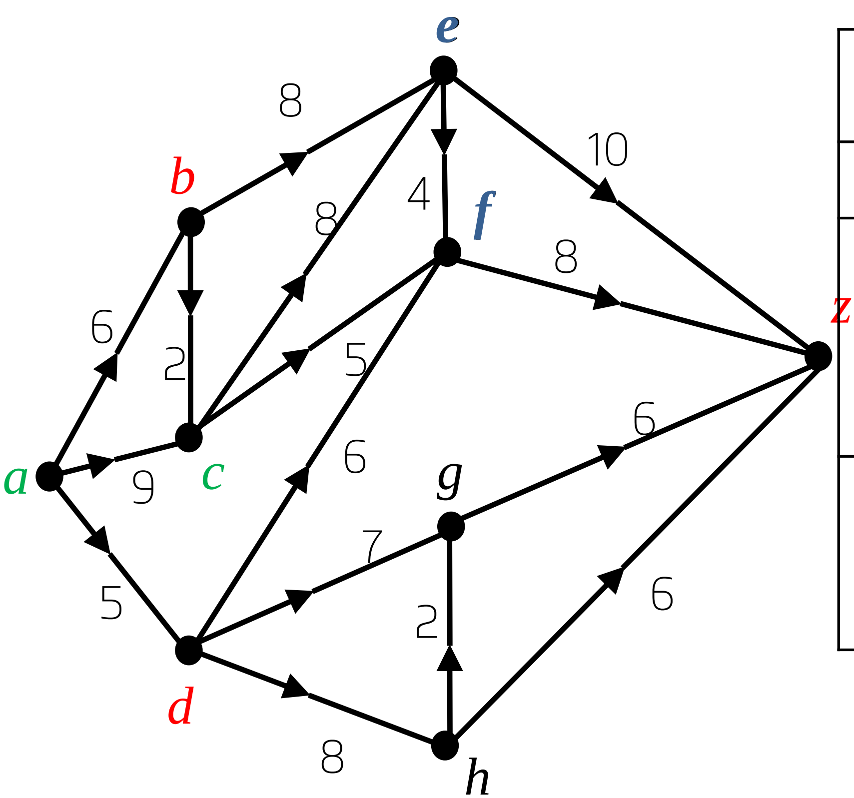
It.	v_d (M)	Mc	A	$v_i,\dots,v_d,v_j$ L(v_j)	X Xd	R
0	---	a	$\{b,c,d\}$	$a,b \rightarrow 6$ $a,c \rightarrow 9$ $a,d \rightarrow 5$	$\{d,b,c\}$ $\{5,6,9\}$	a,b a,c a,d
1	d	a,d	$\{f,g,h\}$	$a,d,f \rightarrow 5+6=11$ $a,d,g \rightarrow 5+7=12$ $a,d,h \rightarrow 5+8=13$	$\{b,c,f,g,h\}$ $\{6,9,11,12,$ $13\}$	a,b a,c a,d,f a,d,g a,d,h

Algoritmo de Dijkstra – Problema do caminho mais curto



It.	v_d (M)	Mc	A	$v_i, \dots, v_d, v_j$ L(v_j)	X Xd	R
0	---	a	$\{b,c,d\}$	a,b → 6 a,c → 9 a,d → 5	$\{d,b,c\}$ $\{5,6,9\}$	a,b a,c a,d
1	d	a,d	$\{f,g,h\}$	a,d,f → 5+6=11 a,d,g → 5+7=12 a,d,h → 5+8=13	$\{b,c,f,g,h\}$ $\{6,9,11,12,13\}$	a,b a,c a,d,f a,d,g a,d,h
2	b	a,b	$\{c,e\}$	a,b,c → 6+2=8 a,b,e → 6+8=14	$\{c,f,g,h,e\}$ $\{8,11,12,13,14\}$	a,b,c a,b,e a,d,f a,d,g a,d,h

Algoritmo de Dijkstra – Problema do caminho mais curto

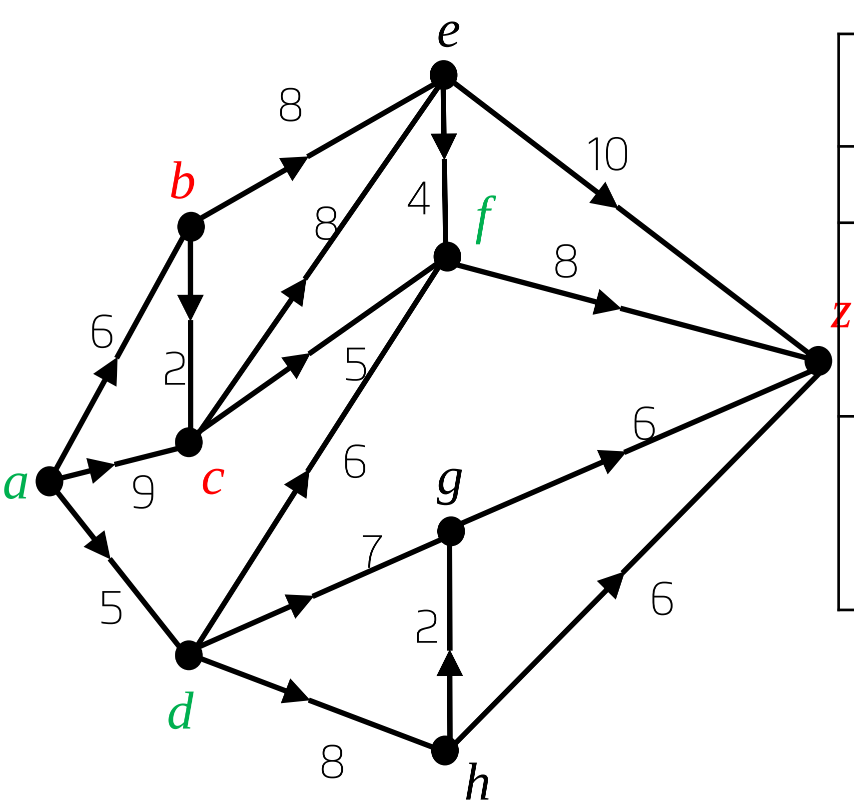


It.	v_d (M)	Mc	A	$v_i, \dots, v_d, v_j$ L(v_j)	X Xd	R
…						
2	b	a,b	$\{c,e\}$	$a,b,c \rightarrow 6+2=8$ $a,b,e \rightarrow 6+8=14$	$\{c,f,g,h,e\}$ $\{8,11,12,13,14\}$	a,b,c a,b,e a,d,f a,d,g a,d,h
3	c	a,c	$\{e,f\}$	$a,c,e \rightarrow 8+8=16$ $a,c,f \rightarrow 8+5=13$	$\{f,g,h,e\}$ $\{11,12,13,14\}$	a,b,e a,d,f a,d,g a,d,h

O caminho a,b,e tem custo 14<16

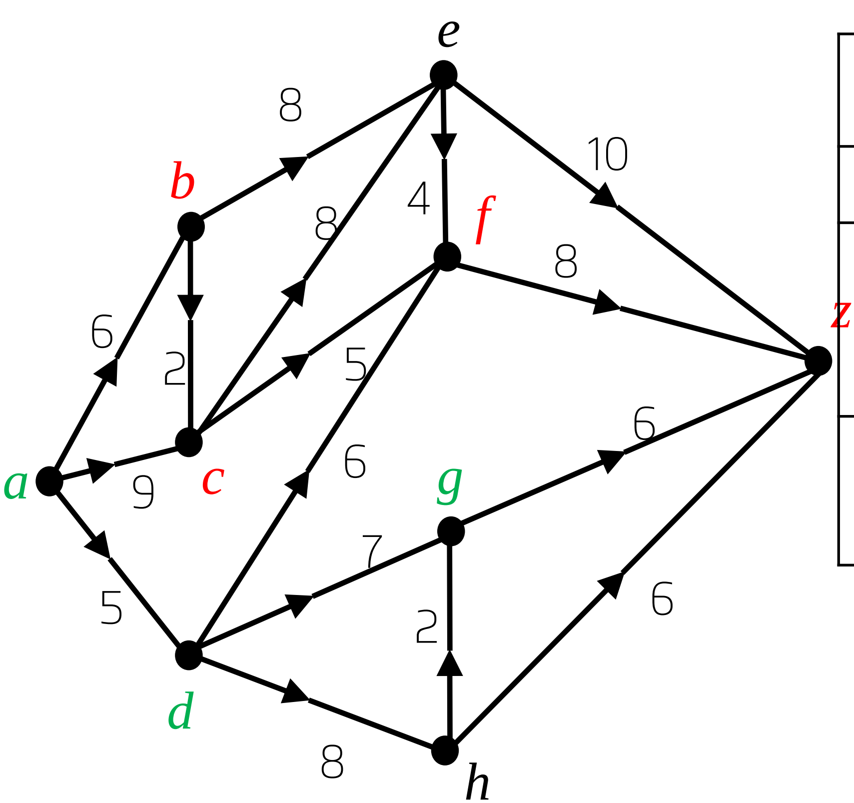
O caminho a,d,f tem custo 11<13

Algoritmo de Dijkstra – Problema do caminho mais curto



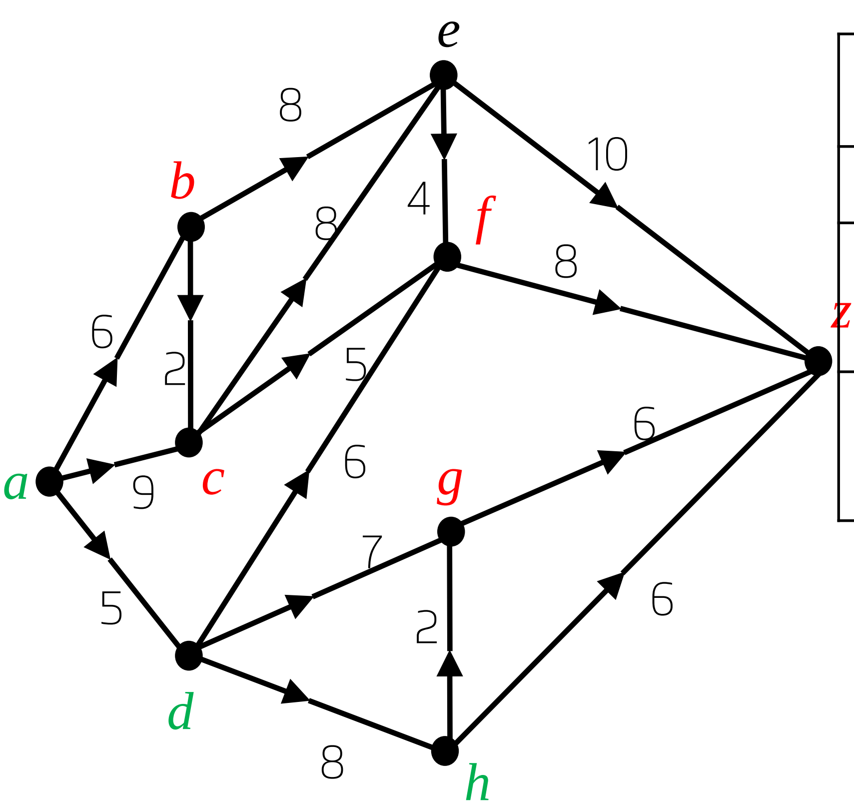
It.	v_d (M)	Mc	A	$v_i, \ldots, v_d, v_j$ L(v_j)	X Xd	R
...						
3	c	a,c	$\{e,f\}$	$a,c,e \rightarrow$ 8+8=16 $a,c,f \rightarrow$ 8+5=13	$\{f,g,h,e\}$ $\{$11, 12, 13,14$\}$	a,b,e a,d,f a,d,g a,d,h
4	f	a,d,f	$\{z\}$	a,d,f,z $\rightarrow$ 11+8=19	$\{g,h,e,z\}$ $\{$12, 13,14,19$\}$	a,b,e a,d,f,z a,d,g a,d,h

Algoritmo de Dijkstra – Problema do caminho mais curto



It.	v_d (M)	Mc	A	$v_i, \dots, v_d, v_j$ L(v_j)	X Xd	R
...						
4	f	a,d,f	$\{z\}$	a,d,f,z →11+8=19	$\{g,h,e,z\}$ {12, 13, 14, 19}	a,b,e a,d,f,z a,d,g a,d,h
5	g	a,d,g	$\{z\}$	a,d,g,z →12+6=18	$\{h,e,z\}$ {13, 14, 18}	a,b,e a,d,g,z a,d,h

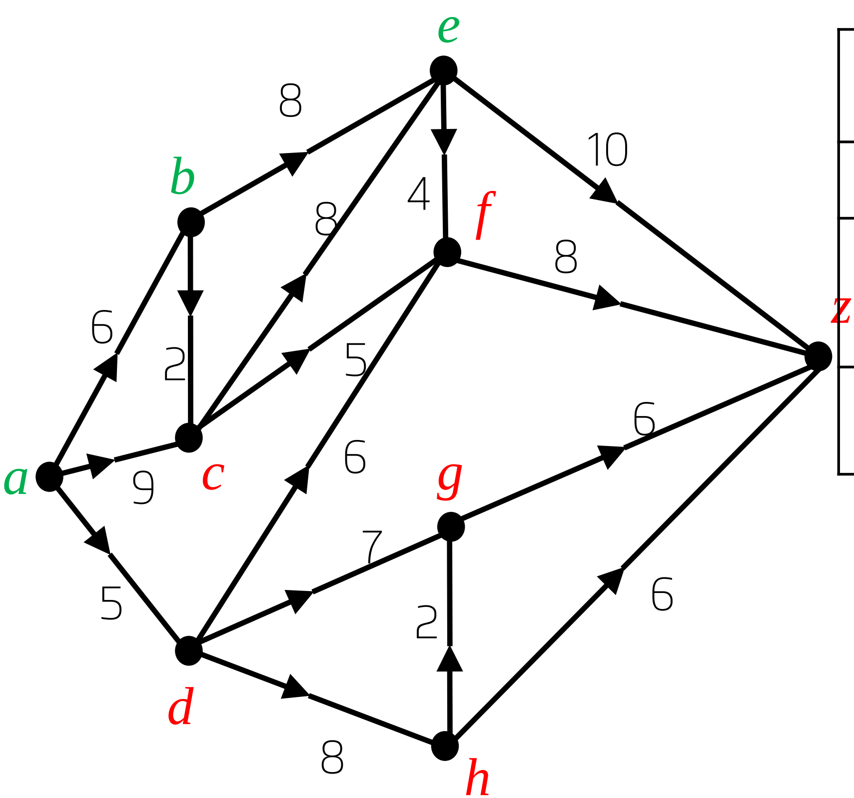
Algoritmo de Dijkstra – Problema do caminho mais curto



It.	v_d (M)	Mc	A	$v_i, \ldots, v_d, v_j$ L(v_j)	X Xd	R
...						
5	g	a,d,g	$\{z\}$	a,d,g,z →12+6=18	$\{h,e,z\}$ $\{13, 14, 18\}$	a,b,e a,d,g,z a,d,h
6	h	a,d,h	$\{z\}$	$\boxed{a,d,h,z}$ →13+6=19	$\{e,z\}$ $\{14, 18\}$	a,b,e a,d,g,z a,d,g

O caminho ***a,d,g,z*** tem custo 18<19

Algoritmo de Dijkstra – Problema do caminho mais curto



It.	v_d (M)	Mc	A	$v_i, \dots, v_d, v_j$ L(v_j)	X Xd	R
…						
6	h	a,d,h	$\{z\}$	a,d,h,z $\rightarrow$13+6=19	$\{e,z\}$ $\{14, 18\}$	a,b,e a,d,g,z a,d,g
7	e	a,b,e	$\{z\}$	$\boxed{a,b,e,\ z}$ $\rightarrow$14+10=24	$\{z\}$ $\{18\}$	a,d,g,z

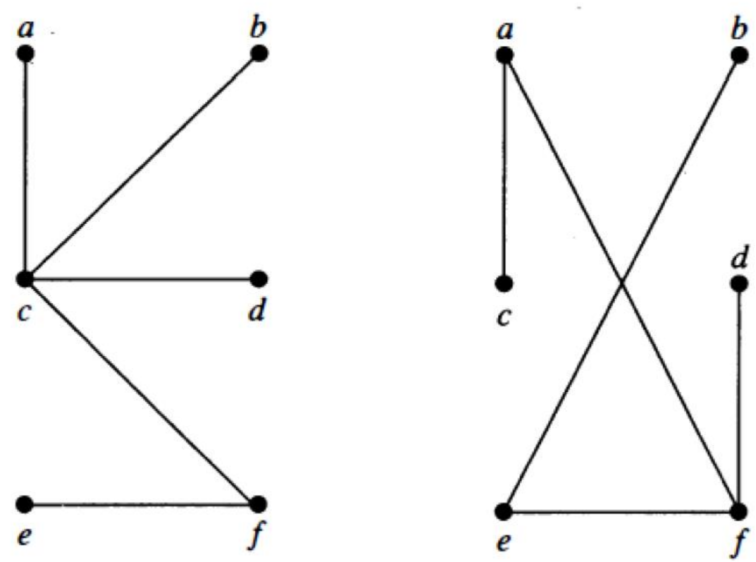
O caminho a,d,g,z tem custo 18 < 24

O caminho mais curto de a a z é a,d,g,z e tem custo 18.

Definição 17:

Árvores (*Trees*) são grafos simples, não orientados, conexos que não contêm circuitos simples. **Florestas** (*Forests*) são grafos não conexos que não contêm circuitos simples, i.e., são coleções não conexas de árvores ou dito de outro modo, são grafos não conexos cujas componentes conexas são árvores.

São árvores



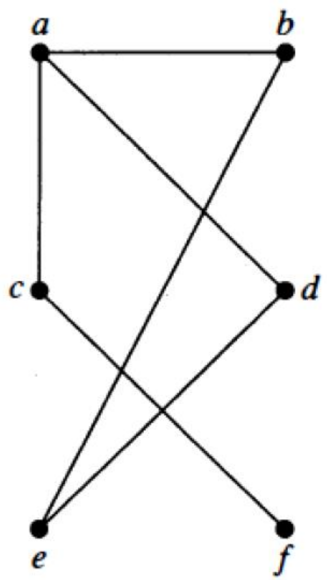
(a) (b)

pois são grafos conexos que não contêm circuitos simples

Definição 17:

Árvores (*Trees*) são grafos simples, não orientados, conexos que não contêm circuitos simples.

Florestas (*Forests*) são grafos não conexos que não contêm circuitos simples, i.e., são coleções não conexas de árvores ou dito de outro modo, são grafos não conexos cujas componentes conexas são árvores.



(c)

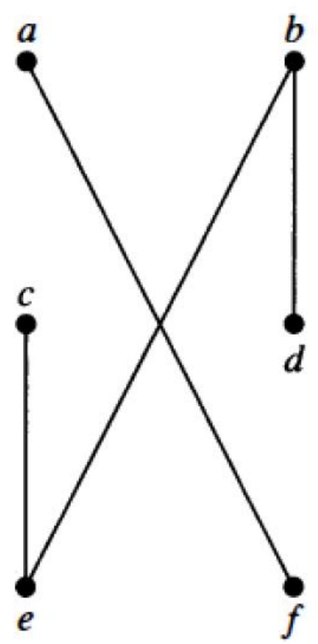
Não é árvore

pois contém, por exemplo, o circuito simples e, b, a, d, e

Definição 17:

Árvores (*Trees*) são grafos simples, não orientados, conexos que não contêm circuitos simples.

Florestas (*Forests*) são grafos não conexos que não contêm circuitos simples, i.e., são coleções não conexas de árvores ou dito de outro modo, são grafos não conexos cujas componentes conexas são árvores.



(d)

Não é árvore

pois não é um grafo conexo.

Por exemplo, não existe um caminho de ***a*** para ***e***

É uma floresta

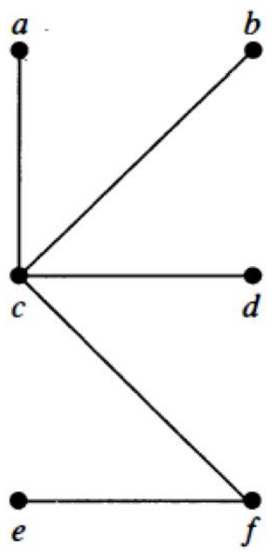
Condições necessárias e suficientes para um grafo ser uma árvore

Teorema 16:

Seja $G = (V, E)$ um grafo simples com v vértices.

Então, G é uma árvore *se e só se* G não contém circuitos simples e tem $v - 1$ arestas. □

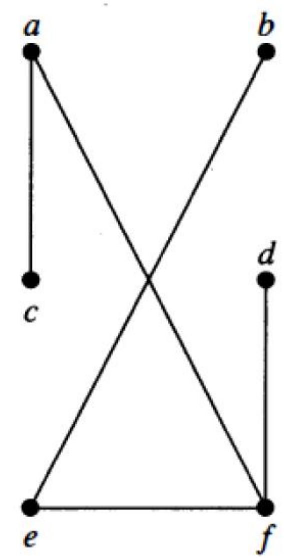
São árvores



(a)

Não contem ciclos

Tem 6 vértices
e
6-1 = 5 arestas



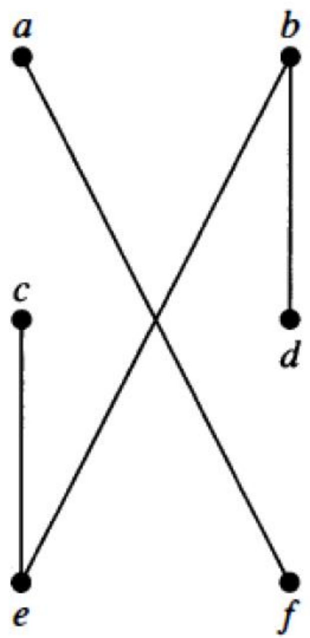
(b)

Condições necessárias e suficientes para um grafo ser uma árvore

Teorema 16:

Seja $G = (V, E)$ um grafo simples com v vértices.

Então, G é uma árvore *se e só se* G não contém circuitos simples e tem $v - 1$ arestas. $\square$



(d)

Não é árvore

Não contem ciclos

Tem 6 vértices
e
4 arestas

Condições necessárias e suficientes para um grafo ser uma árvore

Definição 18:

Designa-se por *vértice de corte* (ou ponto de articulação), um vértice que ao ser removido de um grafo (assim como as arestas nele incidem) produz um subgrafo com um maior número de componentes conexas.

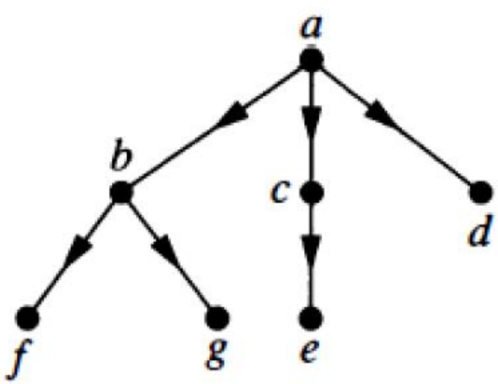
Teorema 17:

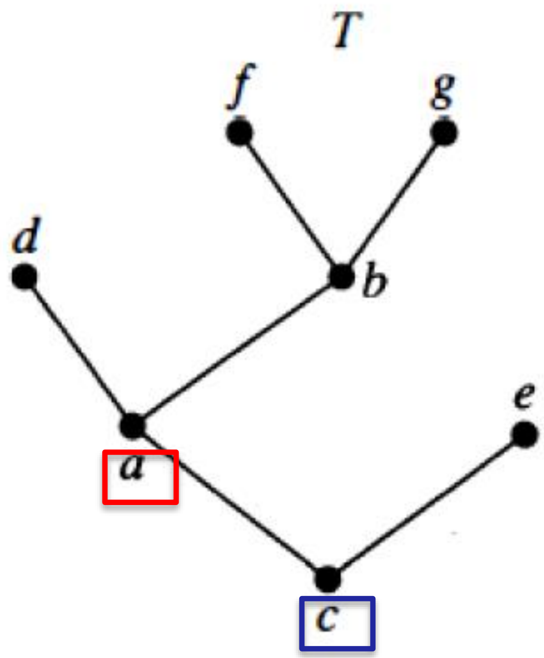
Seja $G = (V, E)$ um grafo simples com v vértices, então são equivalentes as seguintes afirmações:

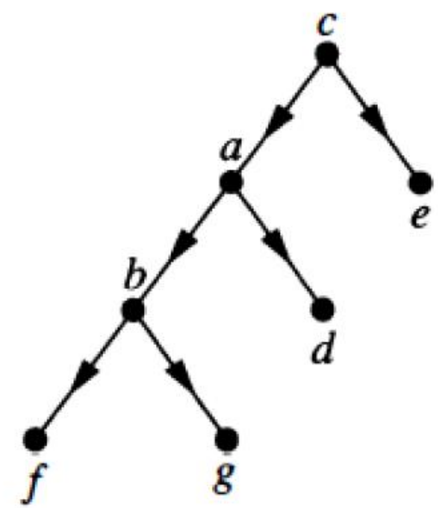
- G é uma árvore;
- G não contém circuitos e tem $v - 1$ arestas;
- G é conexo e tem $v - 1$ arestas;
- G é conexo e cada aresta é uma ponte;
- Existe um único caminho simples entre qualquer par de vértices de G;
- G não contém circuitos, mas acrescentando uma aresta obtém-se um circuito.

Exercício: Verifique quais dos 4 grafos anteriores são árvores.

Exemplo de uma árvore



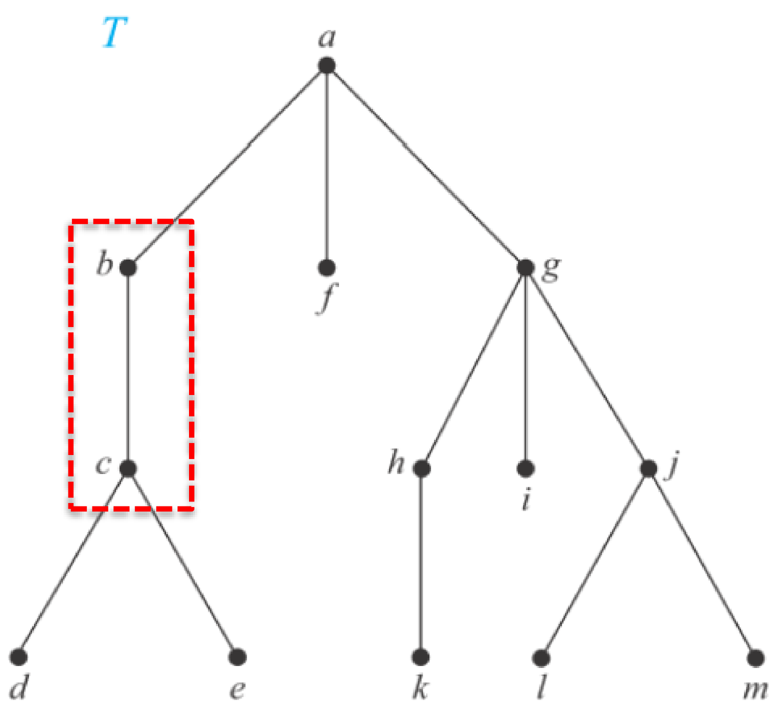




Árvore de raiz a

Árvore de raiz c

Exemplo de uma árvore com raiz a

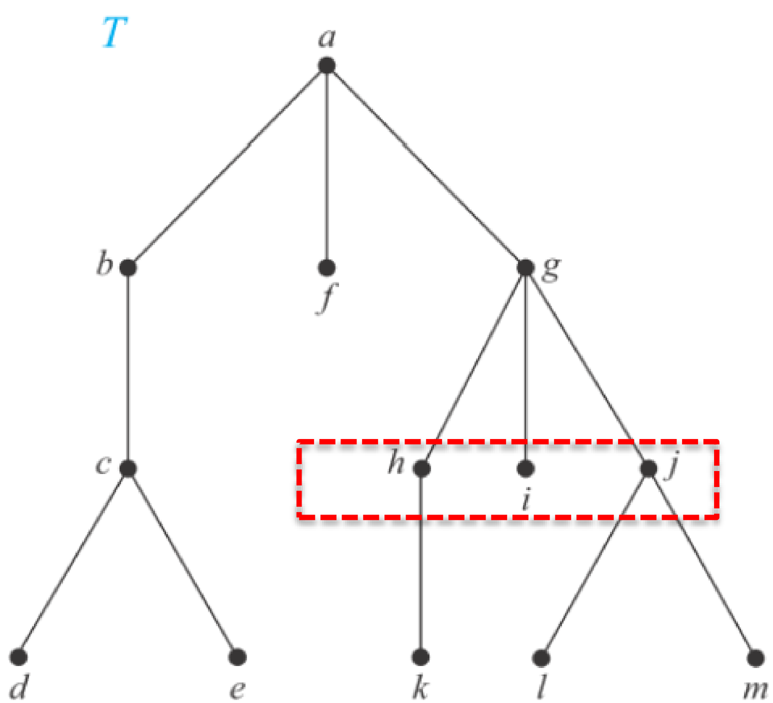


- O pai de c é b.
- Os irmãos de h são i e j.
- Os ascendentes de e são c, b e a.
- Os descendentes de b são c, d e e
- Os vértices internos são a, b, c, g, h e j.
- As folhas são d, e, f, i, k, l e m.

Raiz da árvore

Se v é um vértice diferente de r, o **pai** de v é o único vértice u para o qual existe uma aresta orientada de u para v. Nesse caso, v diz-se um **filho** de u.

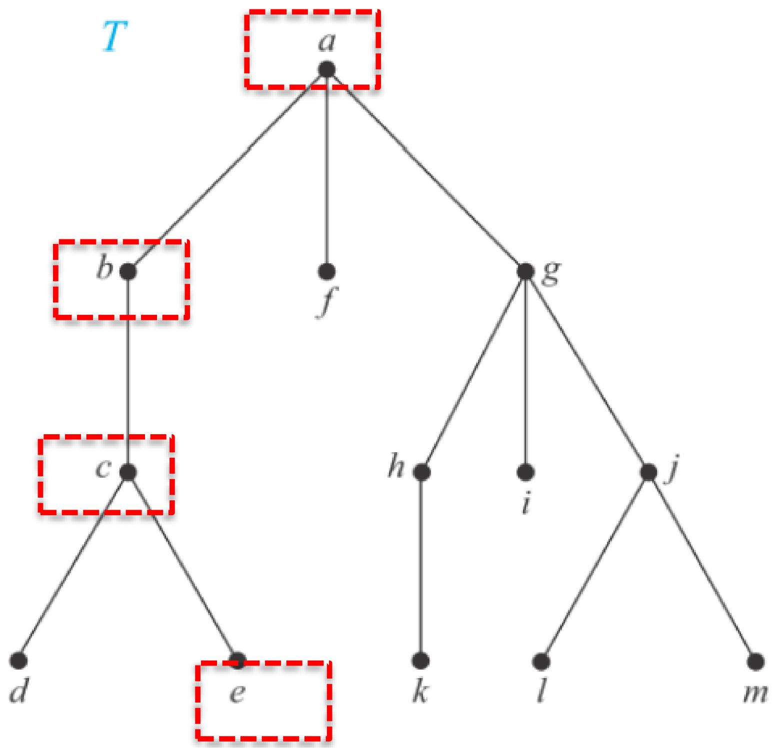
Exemplo de uma árvore com raiz a



- O pai de c é b.
- Os irmãos de h são i e j.
- Os ascendentes de e são c, b e a.
- Os descendentes de b são c, d e e
- Os vértices internos são a, b, c, g, h e j.
- As folhas são d, e, f, i, k, l e m.

Vértices com o mesmo pai são chamados de **irmãos**.

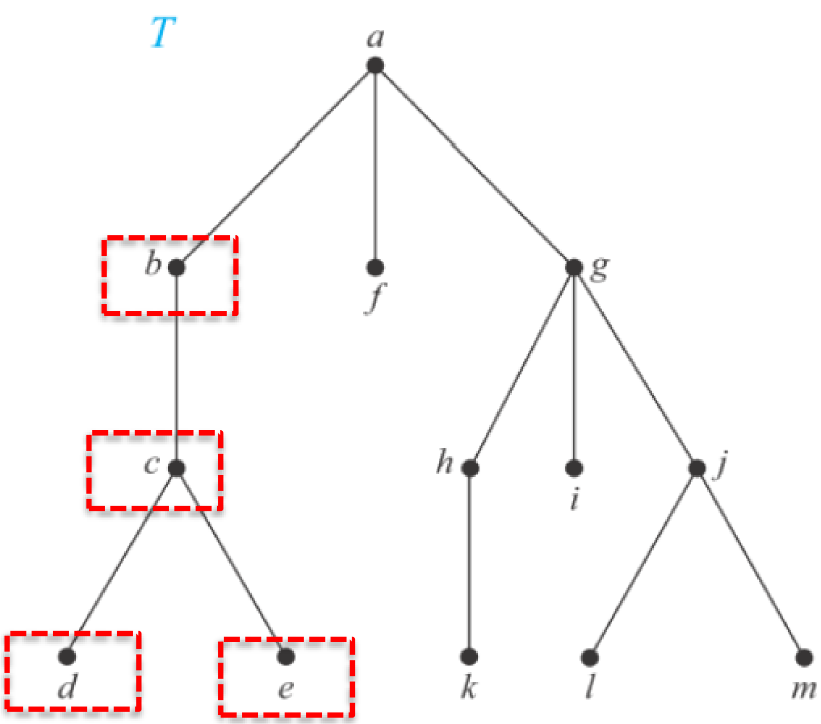
Exemplo de uma árvore com raiz a



- O pai de c é b.
- Os irmãos de h são i e j.
- Os ascendentes de e são c, b e a.
- Os descendentes de b são c, d e e
- Os vértices internos são a, b, c, g, h e j.
- As folhas são d, e, f, i, k, l e m.

Os **ascendentes** de um vértice w, que não a raiz, são os vértices no caminho da raiz a esse vértice, i.e., o pai de w, o pai do seu pai e assim sucessivamente até atingir a raiz.

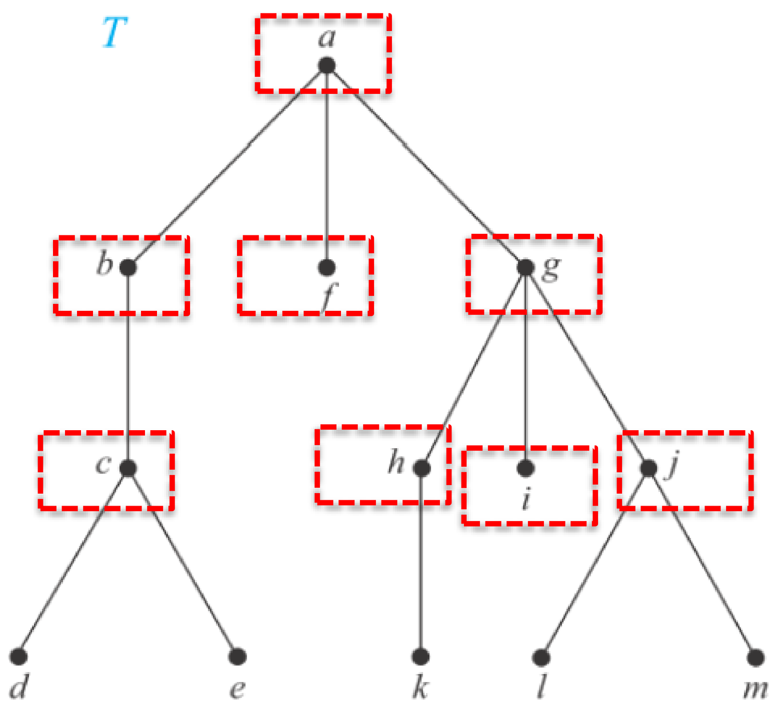
Exemplo de uma árvore com raiz a



- O pai de c é b.
- Os irmãos de h são i e j.
- Os ascendentes de e são c, b e a.
- Os descendentes de b são c, d e e
- Os vértices internos são a, b, c, g, h e j.
- As folhas são d, e, f, i, k, l e m.

Os *descendentes* de um vértice v são todos os vértices que têm como ascendente v.

Exemplo de uma árvore com raiz a

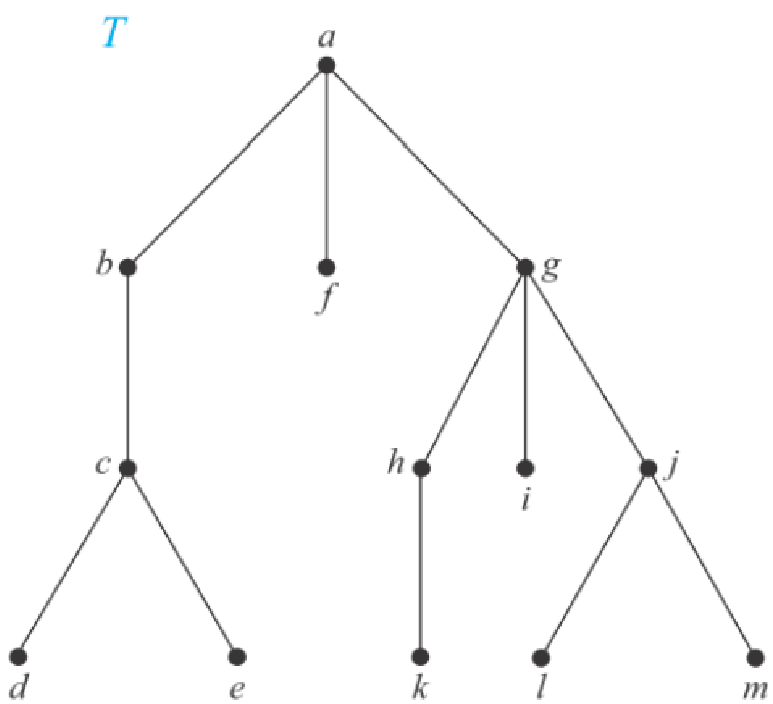


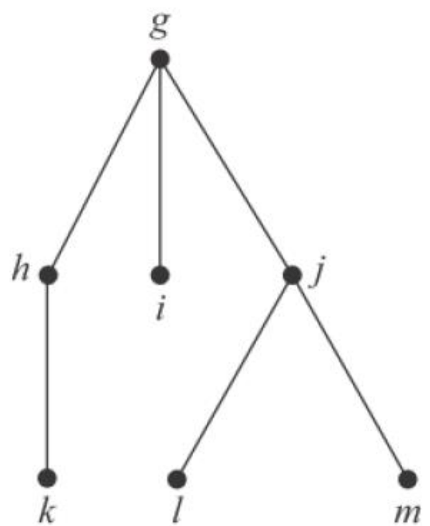
- O pai de c é b.
- Os irmãos de h são i e j.
- Os ascendentes de e são c, b e a.
- Os descendentes de b são c, d e e
- Os vértices internos são a, b, c, g, h e j.
- As folhas são d, e, f, i, k, l e m.

Um vértice de G é uma **folha** se não tiver filhos.

Os vértices que têm filhos dizem-se **vértices internos**.

Exemplo de uma árvore com raiz a

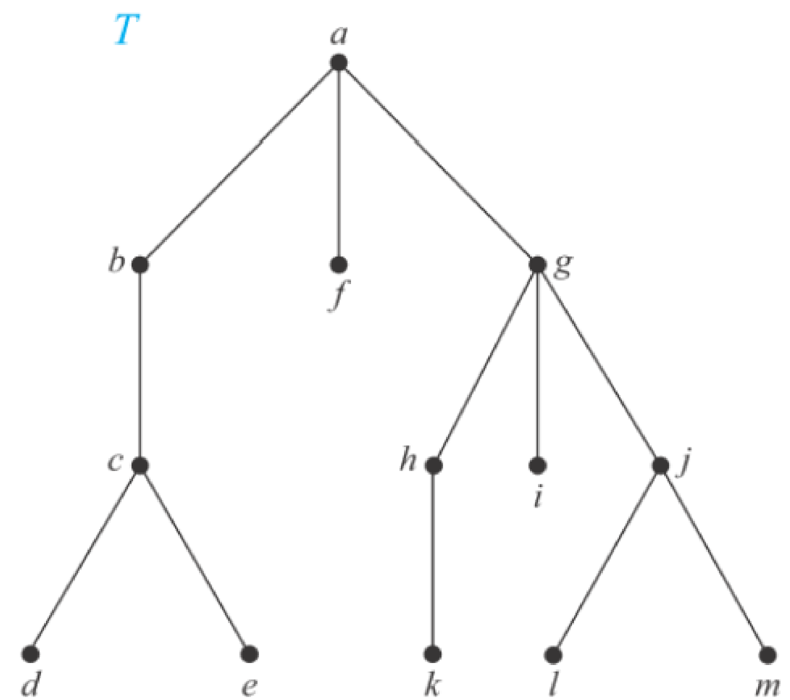




Sub-árvore com raiz g

Se a é um vértice de uma árvore, a **sub-árvore** com raiz a é o subgrafo da árvore que constituído por a e pelos seus descendentes, e todas as arestas incidentes nesses descendentes

- A profundidade de b é
- A profundidade de m é
- A profundidade de h é
- A profundidade de a é
- A altura da árvore T é



A **profundidade do vértice** numa árvore é o comprimento do caminho da raiz até ao vértice.

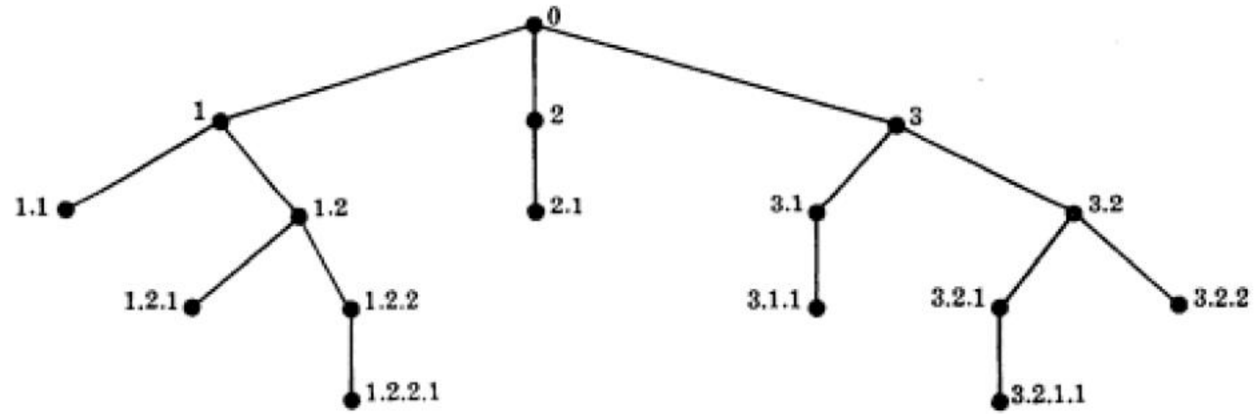
A **altura da árvore** é a maior profundidade de todos os seus vértices, é o comprimento do maior caminho entre a raiz e um vértice.

Definição 19:

Uma árvore com raiz designa-se por **árvore m-ária** se todo o vértice interno não tiver mais de m filhos.

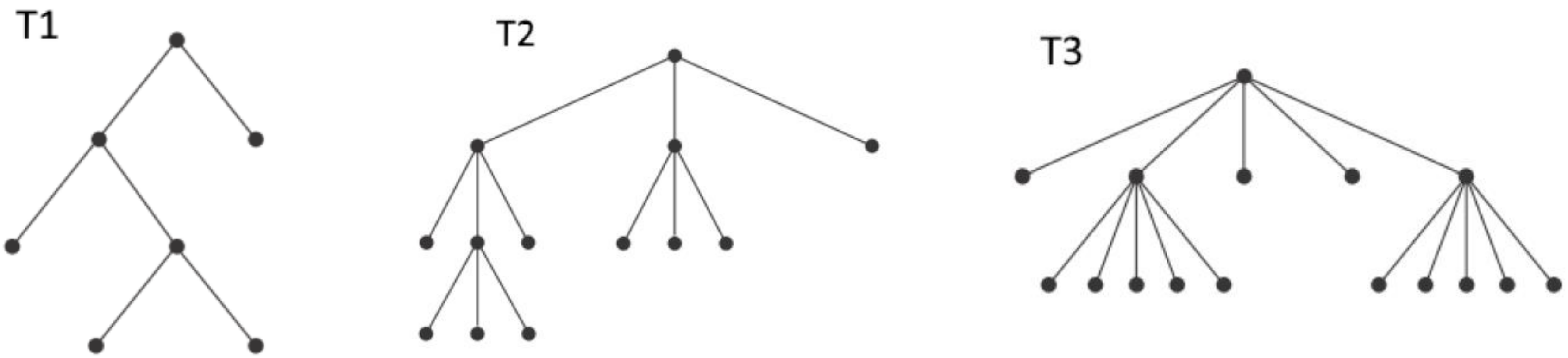
Árvore Binária

Não é
Árvore Binária plena



No caso $m = 2$, a árvore chama-se **binária**

A árvore **árvore m-ária** diz-se uma **árvore m-ária plena** se todo o vértice interno tiver exatamente m filhos.

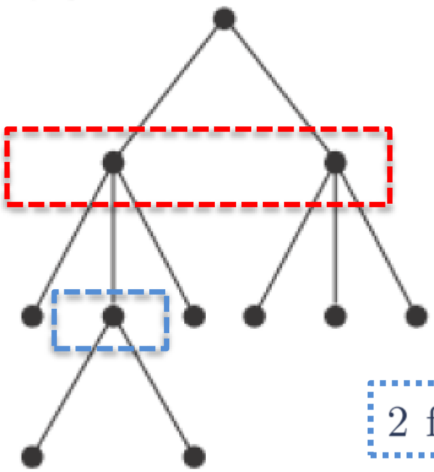


T_1 é uma árvore binária plena porque cada um dos seus vértices internos tem

T_2 é uma árvore porque cada um dos seus vértices internos tem

T_3 é uma árvore porque cada um dos seus vértices internos tem

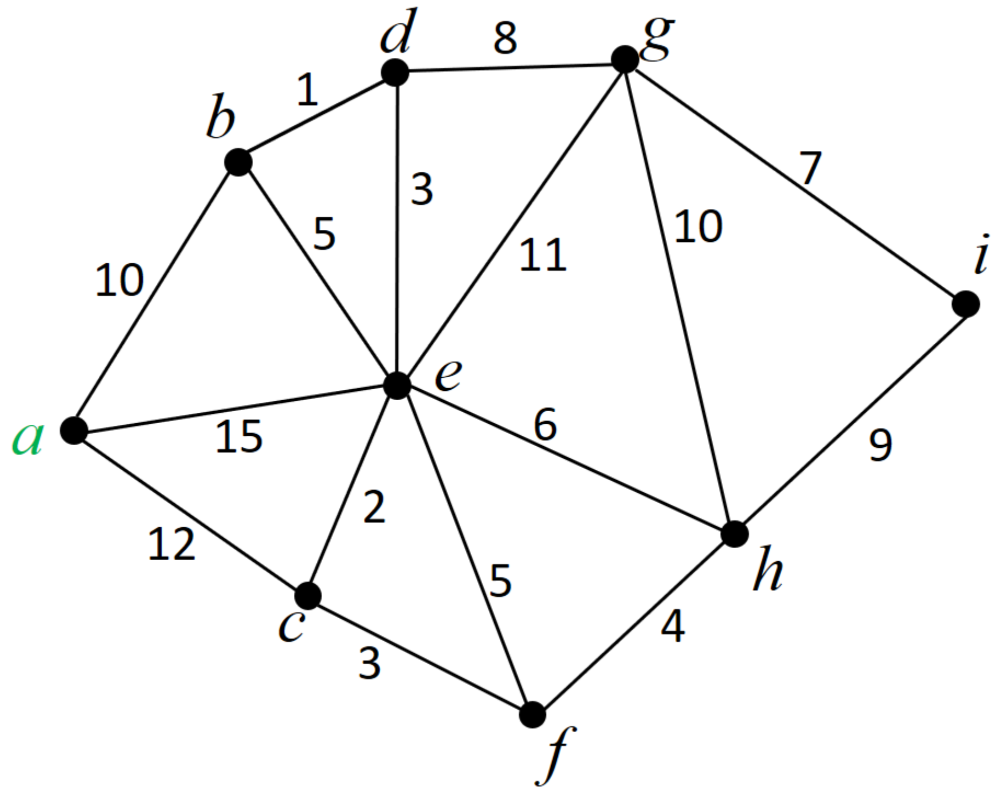
T4



T_4 não é uma árvore m-ária plena porque

têm 3 filhos

2 filhos



Que ligação deve ser estabelecida entre as várias tomadas de casa de forma a usar o menor comprimento de cabo possível?

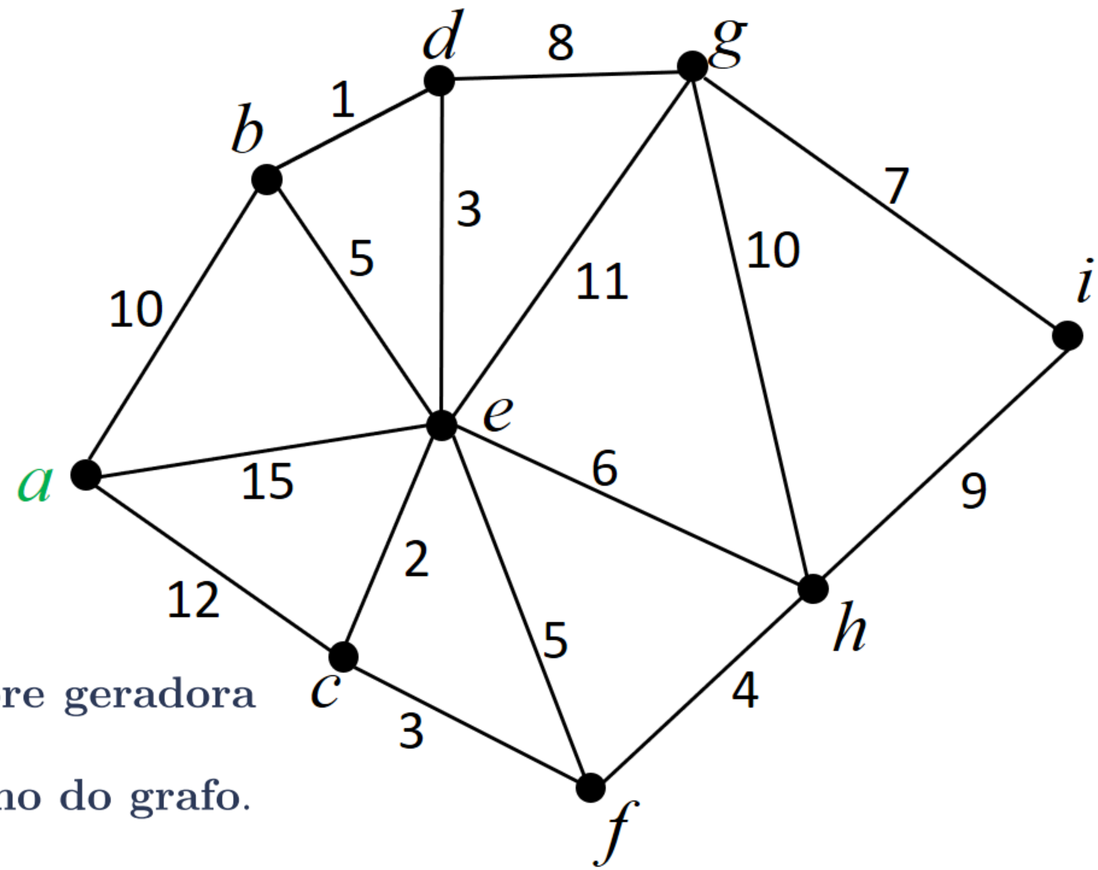
Consideremos um grafo pesado (ou ponderado) no qual:

- os **vértices** representam as tomadas;
- as **arestas** são as ligações entre as tomadas;
- os **pesos** das arestas são os comprimentos dessas ligações.

Árvores geradoras de custo mínimo

Para "levar corrente" de a para g, há várias possibilidades tais como:

- a, b, d, g;
- a, e, g;
- a, e, d, g;
- a, c, e, g;
- entre outras...



Esses 8 caminhos formarão a **árvore geradora de custo mínimo do grafo.**

Do mesmo modo para "levar corrente" de a para cada um dos oito vértices (i.e. b, c, d, e, f, g, h, i terão de ser escolhidos oito caminhos (um por cada vértice). Interessa que a escolha desses caminhos corresponda à utilização do menor comprimento de cabo possível!!!

Definição 20:

Seja G um grafo simples. Uma **árvore geradora** de G é um subgrafo de G que consiste numa árvore que contém todos os vértices de G [a].

Uma **árvore geradora de custo mínimo de um grafo ponderado** é uma árvore constituída por todos os vértices desse grafo e pelas arestas para as quais se obtém um peso total (soma dos pesos dessas arestas) menor.

[a] Para obter árvores geradoras de grafos pode-se usar algoritmos como por exemplo o depth-first search (DFS) ou o breadth-first search (BFS), que podem ser encontrados em [4]

A árvore geradora de custo mínimo de um grafo pode ser obtida usando o **Algoritmo de Kruskal**

Algoritmo de Kruskal

Sejam v_1, v_2, v_n os n vértices do grafo G.
A árvore T será um conjunto de arestas que, no final, constituirão a árvore geradora de custo mínimo.
S será uma partição do conjunto dos vértices de G que irá variar de iteração para iteração.

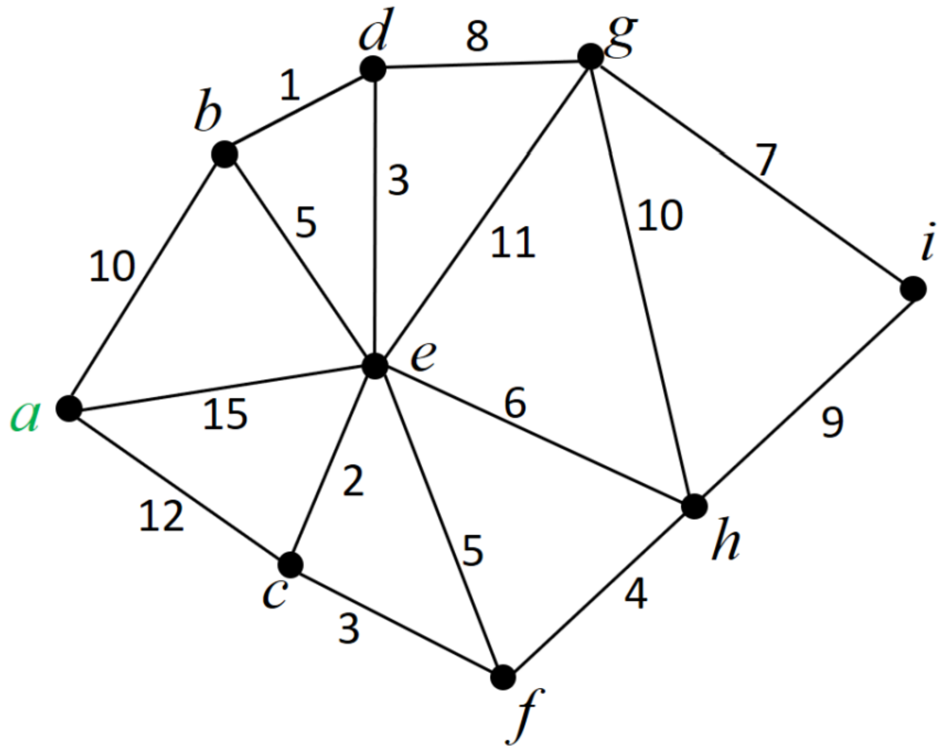
Input: grafo não-orientado conexo com pesos e n vértices

1. **Construir** o vetor RAMO com as arestas por ordem crescente dos seus pesos.
2. Faça-se $T = \emptyset$, $S = \{\{v_1\}, \{v_2\}, \ldots, \{v_n\}\}$, $nr = 0$ e $k = 0$
3. Enquanto $nr < n - 1$

    $k = k + 1$; $(v_i, v_j) = RAMO[k]$;
    Procurar em S o subconjunto de V que contém v_i. Seja S_i esse subconjunto;
    Procurar em S o subconjunto que contém v_j. Seja S_j esse subconjunto;
    Se $S_i \neq S_j$
    então $T = T \cup \{(v_i, v_j)\}$;
        $S_i = S_i \cup S_j$;
        $nr = nr + 1$;

   Fim (Enquanto)

Encontrar a árvore geradora de custo mínimo para o grafo:



Vídeo com a resolução

Resposta: Custo $=38$ e $T = \{(b,d),(c,e),(c,f),(d,e),(f,h),(g,i),(d,g),(a,b)\}$