Índice

1

- Travessia
- Caminho mais curto

Leituras

2

- J. Erickson, Algorithms, Caps. 5, 6.1-6.3, 8, 9.8
- Cormen et al., Introduction to Algorithms, Caps. 22, 24, 25
- Skiena and Revilla, Programming Challenges, Caps. 9
 e 10
- O Dasgupta et al., Algorithms, Cap. 4
- Edmonds, How to think about algorithms, Cap 14

Travessia

3

É comum atravessar um grafo

o à procura de um nó, a etiquetar os nós ou a procurar ciclos, caminho mais curto...

Três tipos comuns de travessia

- Em Largura (Breadth-First Search)
- Em Profundidade (Depth-First Search)
- Por Ordenamento Topológico

Travessia em Largura (BFS)

4

Para um grafo G = (V,E), demora O(|E|+|V|)

```
function BFS(G, v):

Q is an empty queue

Mark v

Enqueue v onto Q

while Q is not empty:

t = dequeue Q

for each edge (t,u) in G

If u is not marked

Mark u
```

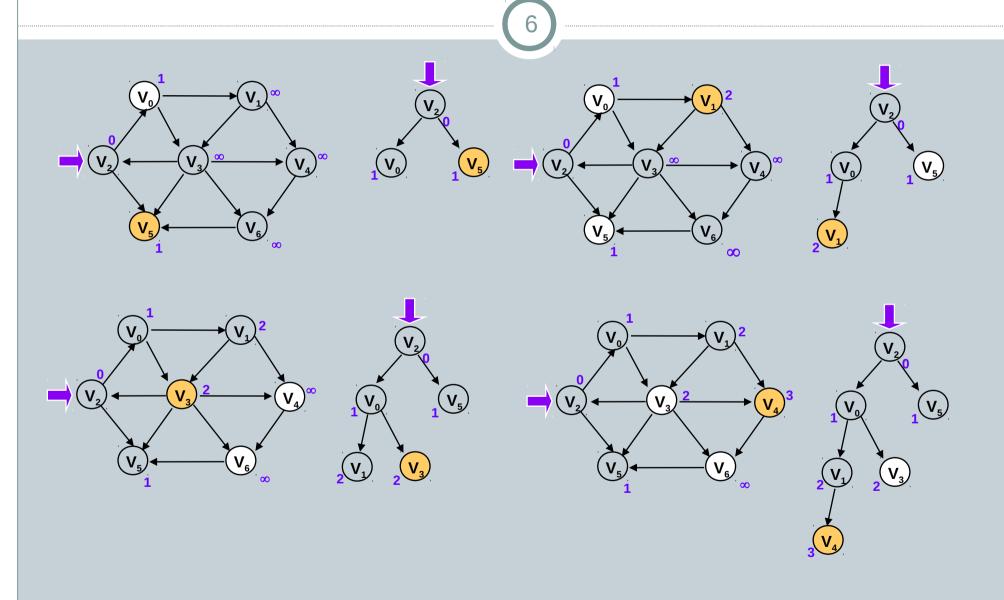
Enqeue u onto Q

Travessia em Largura (BFS)

 Exemplo: etiquetar os nós por distância ao nó inicial, grafo não ponderado

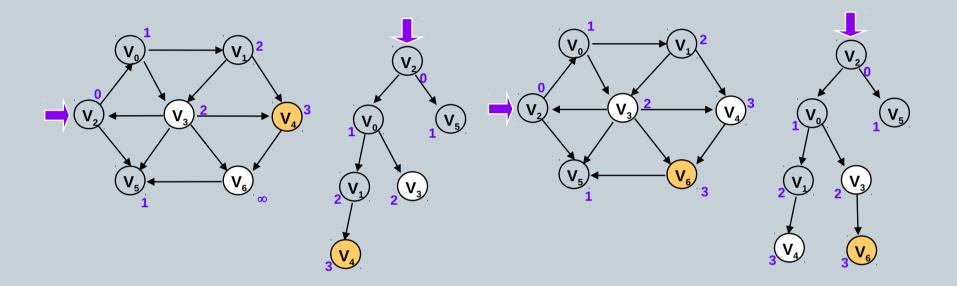
Explora os nós por camadas em que: $\begin{array}{c} & & & \\ & &$

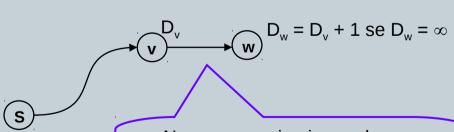
BFS (cont.)



BFS (cont.)







Na procura primeiro em largura também fica claro que v é o nó que precede w no percurso mais curto

Travessia em Profundidade (DFS)

8

Para um grafo G = (V,E), demora O(|E|+|V|)

```
function DFS(G, v):

S is an empty stack

Push v onto S

while S is not empty:

pop t from S

if t is not marked

Mark t

for each edge (t,u) in G

push u onto S
```

Travessia em Profundidade (DFS)

9

function DFS(G, v):

Mark v
for each edge (v,u) in G
If u is not marked
DFS(G,u)

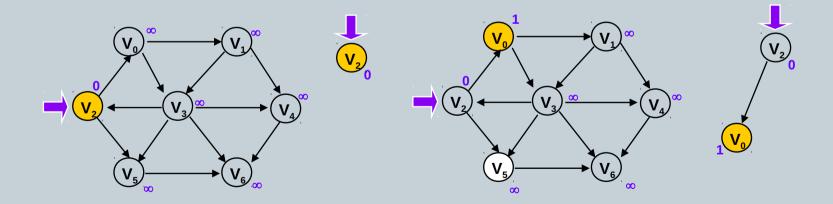
Para um grafo G = (V,E), demora O(|E| + |V|)

Travessia em Profundidade (DFS)

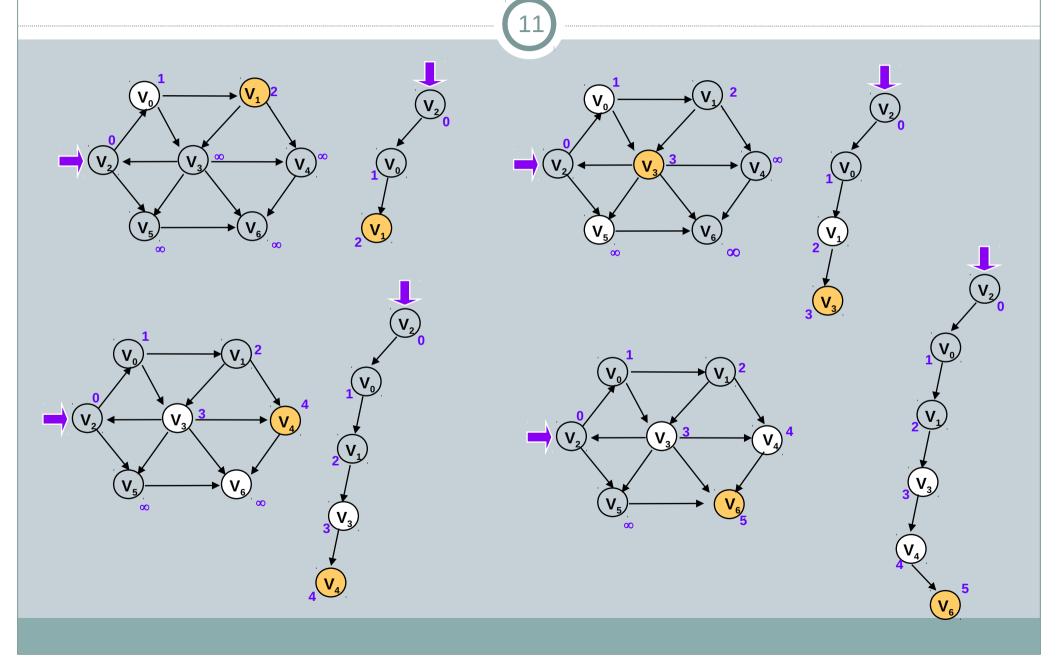
10

• Exemplo: etiquetar os nós por ordem de visita

Em caso de ambiguidade, dar preferência a índice mais baixo

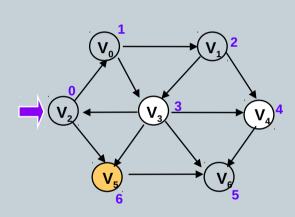


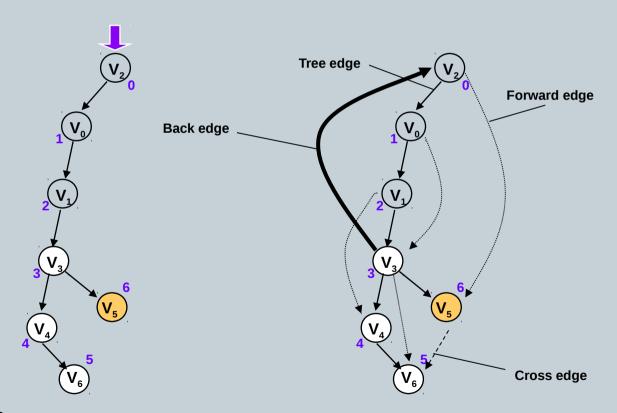
DFS (cont.)



DFS (cont.)

12





Nota: podiamos ter também determinado a distância mais curta com verificação de nós já visitados

Travessia em Ordenação Topológica

Ordenação Topológica

Numa ordenação topológica os nós de um grafo dirigido **acíclico** estão ordenados de tal forma que se existir um caminho de u para v, u precede v no grafo.

Ex.: grafo usado para representar as precedências das disciplinas de um curso. Um arco (v, w) representa que a disciplina v deve estar concluída antes de frequentar a disciplina w.

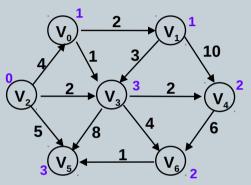
Um grafo pode ter várias ordenações topológicas

Ordenação Topológica

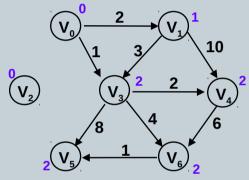


- Calcular o grau interno de todos os nós
- Enquanto existirem nós no grafo com grau interno 0 Faz
 - Seleccionar um nó v cujo grau interno seja 0
 - Processar o nó v
 - Remover o nó v do grafo juntamente com os arcos que saem desse nó
 - Actualizar o grau interno dos nós adjacentes a v
- Fim Enquanto

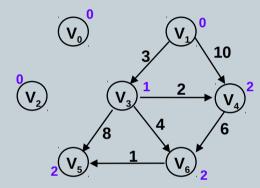
Ordenação Topológica - Exemplo



Ordem topolog.= []



Ordem topolog.= $[V_2]$



Ordem topolog.= $[V_2V_0]$



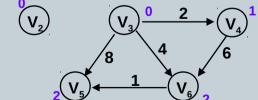








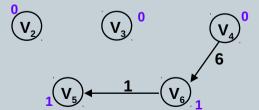




Ordem topolog.= $[V_2V_0V_1]$

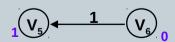






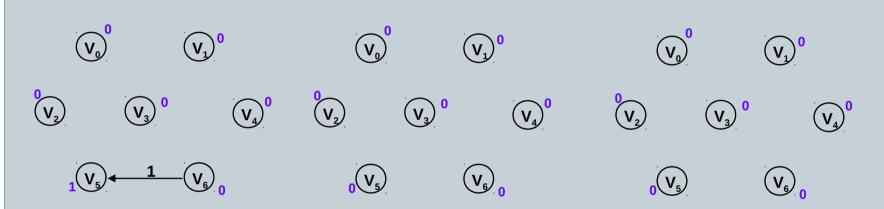
Ordem topolog.=
$$[V_2V_0V_1V_3]$$





Ordem topolog.= $[V_2V_0V_1V_3V_4]$

Ordenação Topológica (Cont.)



Ordem topolog.= $[V_2V_0V_1V_3V_4]$ Ordem topolog.= $[V_2V_0V_1V_3V_4V_6]$ Ordem topolog.= $[V_2V_0V_1V_3V_4V_6V_5]$

Travessias BFS, DFS e ord. top. têm complexidade linear...



- Podemos utilizar o algoritmo BFS para determinar o caminho mais curto
- o ...mas só em condições muito restritas
 - Sem pesos ou com pesos unitários (BFS)
 - Sem ciclos (Ord. Topológica)

18

Algoritmo de Dijkstra

- Permite fazer uma pesquisa mais rápida (grafos de maiores dimensões)
- Permite ciclos
- ...não permite pesos negativos

Algoritmo de Bellman-Ford

- Permite pesos negativos
- Detecta ciclos
- ...é lento para grafos de grande dimensão

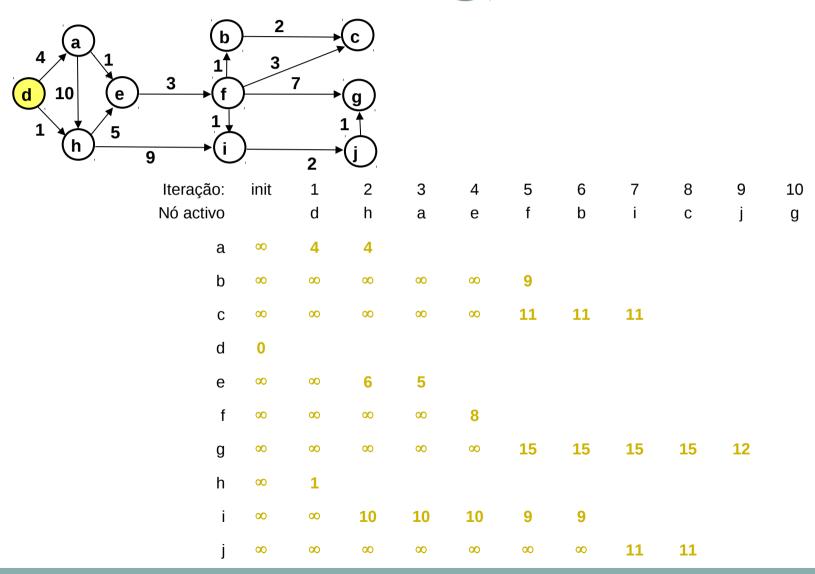


- 1. Inicializar estimativas das distâncias nos nós:
 - nó fonte com 0
 - restantes nós com infinito;
 - Marcar todos os nós como *não-visitados*, exceto o nó fonte, que fica marcado como *atual*.
- 2. Considerar todos nós não-visitados que são vizinhos do atual e atualizar as estimativas: a menor entre a distância estimada ao nó não-visitado e a distância estimada pelo caminho que passa pelo nó atual; marcar o nó atual como visitado.
- 3. Parar se o nó destino estiver marcado como visitado; caso contrário, selecionar o nó com a menor distância estimada e definir como atual e goto 2.

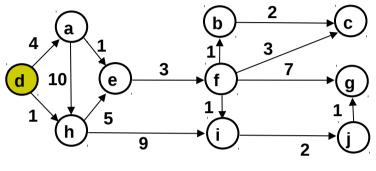


```
function Dijkstra(G, source, target):
for each vertex v in G
    dist[v] = infinity
dist[source] = 0
Q has the set of all nodes in G
while Q is not empty:
    u = vertex in Q with smallest dist
    remove u from Q
    if u = target
       break
    for each arc (v,u) in G
        if dist[v] > dist[u] + dist between(v, u)
          dist[v] = dist[u] + dist_between(v, u)
return dist
```









 ∞

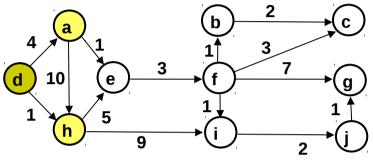
00

 ∞

```
Iteração:
                  init
                                        2
                                                                                                                             10
Nó activo
                                        h
                              d
                                                                                                                             g
           a
                             \infty
                                        \infty
                                                   \infty
                                                                       11
                                                                                  11
                                                                                             11
                   0
                                                   5
                             \infty
                   \infty
                             \infty
                                        \infty
                                                                                                       15
                                                                                                                 12
                             \infty
                              1
                                        10
                                                  10
                   \infty
                             \infty
                                                             10
                                                                                             11
                                                                                                       11
```

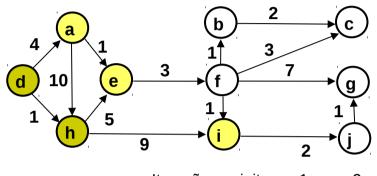
 ∞





```
Iteração:
                  init
                                          2
                                                                                                                                 10
                               1
Nó activo
                                          h
                               d
                                                                                                                                  g
                                          4
                                                                          11
                                                                                     11
                                                                                                11
                    \infty
                               \infty
                    0
                                                     5
                    \infty
                               \infty
                               \infty
                                                                          15
                                                                                     15
                                                                                                           15
                                                                                                                      12
                               \infty
                                          00
                                                     00
                               1
                    \infty
                                         10
                                                    10
                                                               10
                    \infty
                               \infty
                                                                                                           11
                                                                                                11
                    \infty
                               \infty
                                          \infty
                                                     \infty
                                                                           00
```

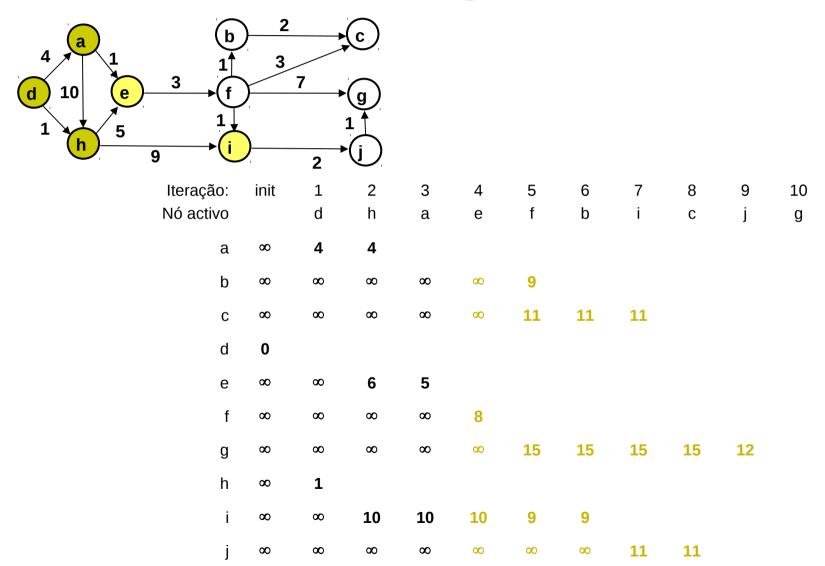




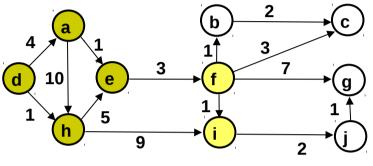
```
Iteração:
                                           2
                   init
                               1
                                                                                                                                   10
Nó activo
                                d
                                          h
            a
                               \infty
                                          \infty
                                                      \infty
                                                                           11
                                                                                      11
                                                                                                  11
                    0
                    \infty
                               \infty
                    \infty
                               \infty
                                                                           15
                                                                                                             15
                                                                                                                        12
                               \infty
                                1
                                          10
                               \infty
                                                     10
                                                                10
                    \infty
                                                                                                  11
                                                                                                             11
                               \infty
                                           \infty
                    \infty
```

g



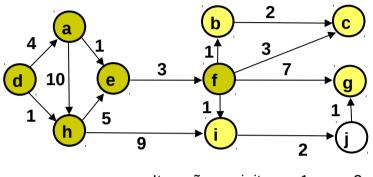






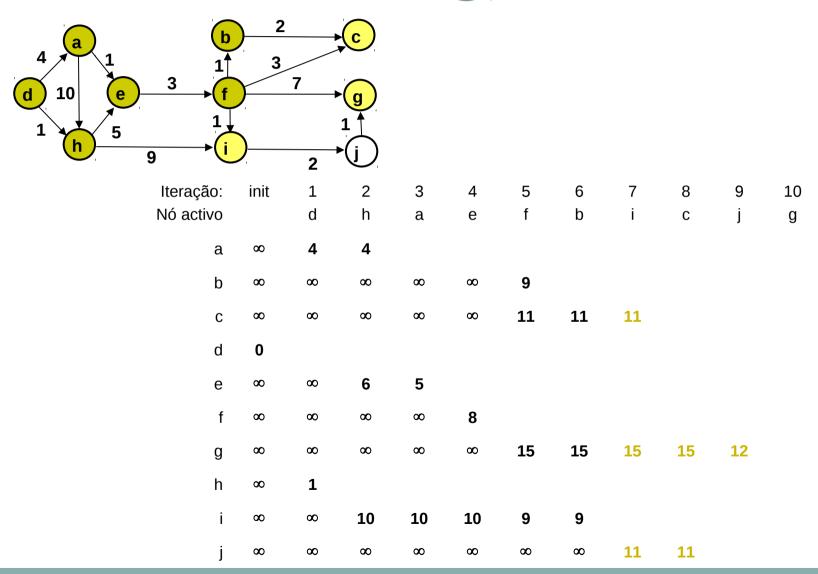
```
Iteração:
                  init
                                          2
                                                                                                                                10
Nó activo
                                         h
                               d
                                                                                                                                 g
                              \infty
                                         \infty
                                                                         11
                                                                                    11
                                                                                               11
                    0
                                                    5
                              \infty
                    \infty
                              \infty
                                                                         15
                                                                                    15
                                                                                                          15
                                                                                                                     12
                              \infty
                               1
                                         10
                   \infty
                              \infty
                                                    10
                                                               10
                                                                                               11
                                                                                                          11
                                         \infty
                    \infty
                               \infty
                                                               \infty
                                                                          00
```



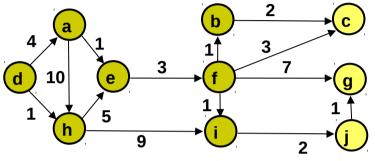


```
Iteração:
                   init
                               1
                                          2
                                                                                                                                 10
                                                     3
Nó activo
                                         h
                               d
                                                     a
                                                                          11
                                                                                     11
                                                                                                11
                               \infty
                                          \infty
                                                     \infty
                                                                \infty
                    0
                                                     5
                               \infty
                                                                          15
                                                                                     15
                                                                                                                      12
                               \infty
                               1
                                         10
                                                    10
                                                               10
                               \infty
                                                                                                11
                                                                                                           11
                    \infty
                               \infty
                                          \infty
                                                     \infty
                                                                \infty
                                                                           \infty
```



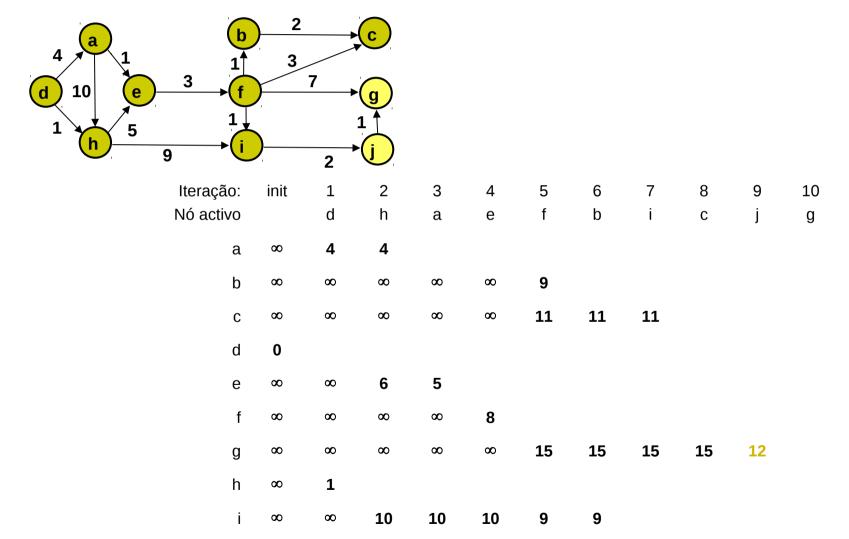






```
Iteração:
                   init
                                            2
                                1
                                                                                                                                        10
Nó activo
                                            h
                                d
                                                        а
                                                                                                                                         g
                     \infty
                                \infty
                                                                              11
                                                                                         11
                                                                                                     11
                     \infty
                                \infty
                                                        5
                     \infty
                                \infty
                                                                              15
                                                                                         15
                                                                                                     15
                                                                                                                15
                                                                                                                            12
                     \infty
                                \infty
                                1
                     \infty
                                           10
                                                       10
                                                                  10
                                \infty
                     \infty
                                \infty
                                            \infty
                                                                   \infty
                                                                               \infty
                                                                                                     11
                                                                                                                11
```





 ∞

 ∞

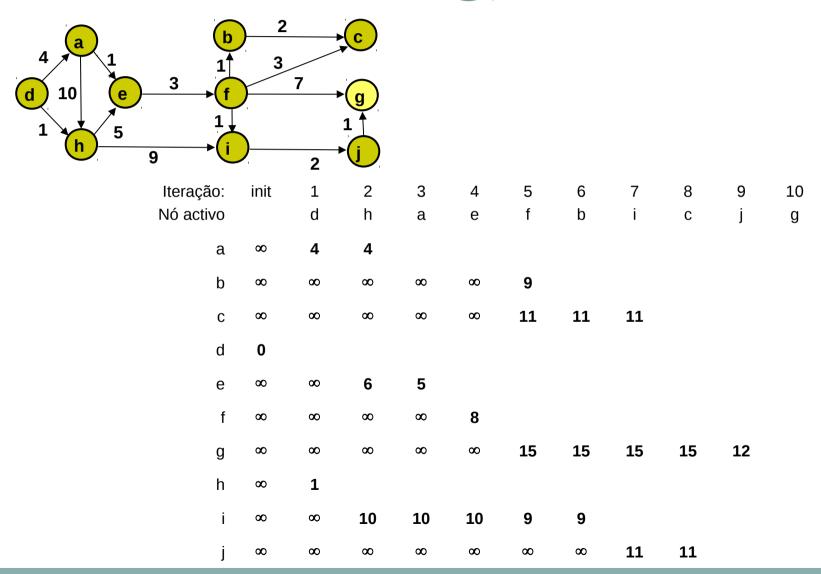
 ∞

11

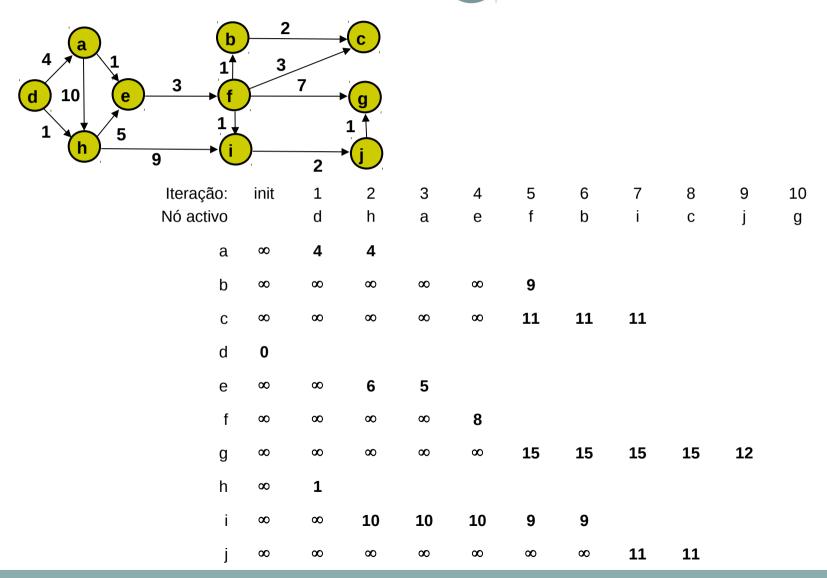
11

 ∞

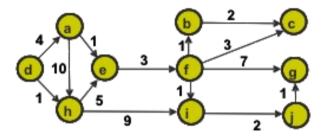












Iteração:	init	1	2	3	4	5	6	7	8	9	10
Nó activo		d	h	a	e	f	b	i	С	j	g
а	00	4	4								
b	00	00	00	00	00	9					
С	00	200	00	00	00	11	11	11			
d	0										
е	00	200	6	5							
f	8	00	00	00	8						
g	00	20	00	00	00	15	15	15	15	12	
h	8	1									
i	00	200	10	10	10	9	9				
j	80	00	00	00	00	00	80	11	11		

PROBLEMA.: o algoritmo de dijkstra falha quando temos pesos negativos !!

34

```
function Dijkstra(G, source, target):
                                                 Para um grafo G = (V,A),
for each vertex v in G
                                                 demora O(|V|^2) se Q é uma
    dist[v] = infinity
                                                 lista ligada
dist[source] = 0
O has the set of all nodes in G
while Q is not empty:
                                                   \leftarrow O(|V|)
    u = vertex in Q with smallest dist
                                                   \leftarrow O(|V|)
                                                   \leftarrow O(1)
    remove u from Q
    if u = target
       break
     for each arc (v,u) in G
                                                   \leftarrow O(|A|) - for all arcs
        if dist[v] > dist[u] + dist between(v, u)
          dist[v] = dist[u] + dist_between(v, u)
return dist
```



```
Para um grafo G = (V,A),
function Dijkstra(G, source, target):
                                               demora O((|A| + |V|) \log |V|)
for each vertex v in G
                                               se Q é uma árvore binária
    dist[v] = infinity
                                               balanceada
dist[source] = 0
O has the set of all nodes in G
while Q is not empty:
                                                  \leftarrow O(|V|)
    u = vertex in Q with smallest dist
                                                  \leftarrow O(\log |V|)
                                                  \leftarrow O(\log |V|)
    remove u from O
    if u = target
       break
                                                  \leftarrow O(|A|) - for all arcs
     for each arc (v,u) in G
       if dist[v] > dist[u] + dist between(v, u)
          dist[v] = dist[u] + dist_between(v, u)
                                                  \leftarrow O(\log |V|)
          update Q
return dist
```

36

```
Para um grafo G = (V,A),
function Dijkstra(G, source, target):
                                               demora O(|A| + |V| \log |V|)
for each vertex v in G
                                               se Q é uma fibonacci heap
   dist[v] = infinity
dist[source] = 0
O has the set of all nodes in G
while Q is not empty:
                                                 \leftarrow O(|V|)
                                                 \leftarrow O(1)
    u = vertex in Q with smallest dist
    remove u from O
                                                 \leftarrow O(\log |V|) - tempo amortizado
    if u = target
       break
                                                 \leftarrow O(|A|) - for all arcs
     for each arc (v,u) in G
       if dist[v] > dist[u] + dist between(v, u)
          dist[v] = dist[u] + dist_between(v, u)
          update Q
                                                 \leftarrow O(1) - tempo amortizado
return dist
```

37

- O algoritmo de Bellman-Ford permite determinar o caminho mais curto em grafos ponderados com pesos negativos
- Permite também determinar a existência de ciclos negativos

38

Princípio do "relaxamento":

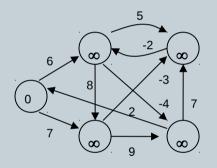
- Sempre que encontramos um vértice v, ligado a
 u, tal que d[v]>d[u]+dist(u,v) então
 - d[v] = d[u] + dist(u,v)
- Os outros algoritmos estudados aplicam esta regra apenas uma vez a cada arco (e.g. Dijkstra)
- O algoritmo Bellman-Ford aplica |V|-1 vezes a cada arco – É o número máximo de vezes para obter convergência na distância estimada quando não existe ciclos negativos.

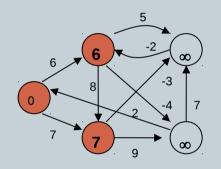
39

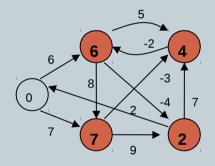
```
Para um grafo G = (V,A),
                                               demora O(|A||V|)
function BelmannFord(G, source):
for each vertex v in G
    dist[v] = infinity
d[source]=0;
for(i=0; i<|V|-1; i++)
                                              \leftarrow O(|V|)
                                              \leftarrow O(|A|)
   for each arc (u,v) in G
         if dist[v] > dist[u] + dist between(u, v)
                  d[v] = d[u] + dist between(u, v)
for each arc (u,v) in G
                                              Verificação de ciclos negativos
   if dist[v] > dist[u] + dist between(u, v)
         return false
                                              Ciclo negativo!!
return true
```

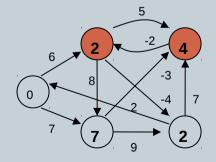


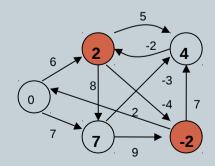
• Exemplo (N=5; M=10)







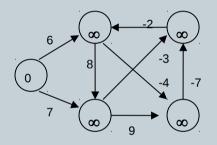


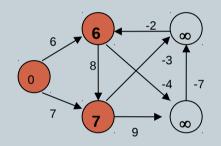


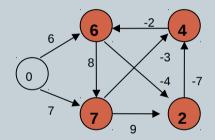
Não há ciclos de peso negativo!

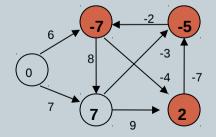
41

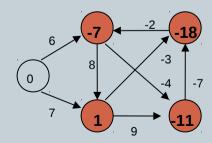
Outro exemplo (N=5; M=8)











Há um ciclo negativo!