Centro Universitário Presidente Antônio Carlos
Teoria de Grafos

Caminho Mínimo - Algoritmo Dijkstra

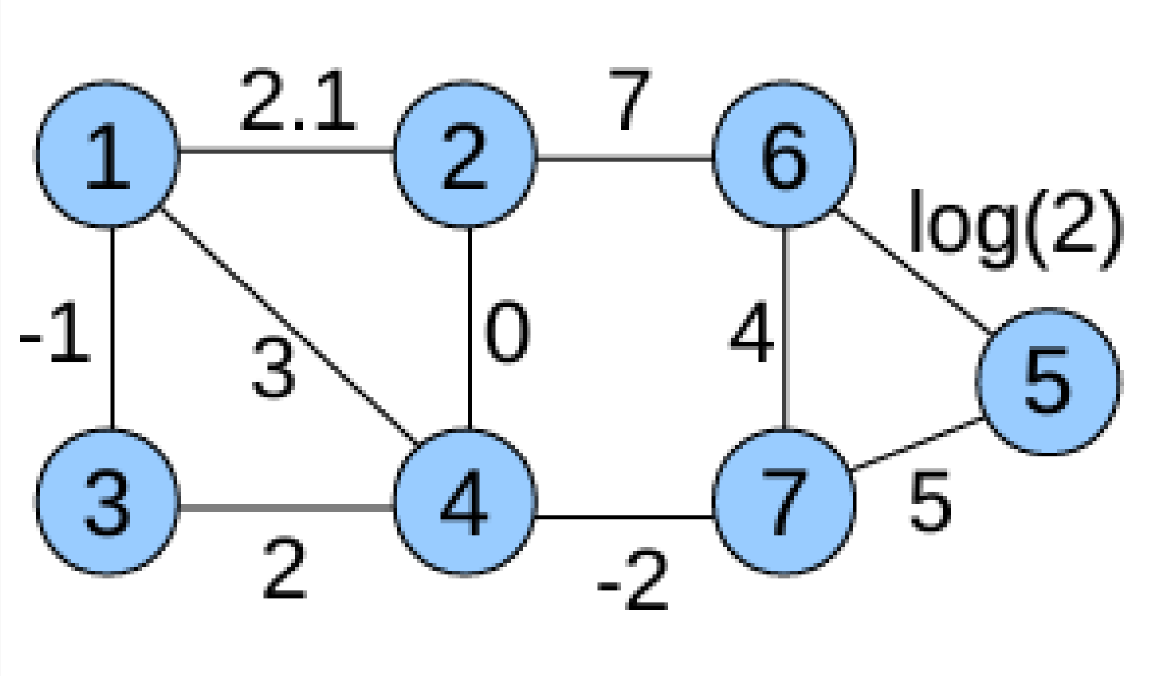
Felipe Roncalli de Paula Carneiro
felipecarneiro@unipac.br

O que vamos aprender nessa aula

- Grafos com pesos;
- Caminhos Mínimos
- Dijkstra;
- Algoritmo de Dijkstra;

Grafos com Pesos

Anotar arestas do grafo com “intensidade” do relacionamento peso da aresta (weight) função w(e) retorna peso da aresta:



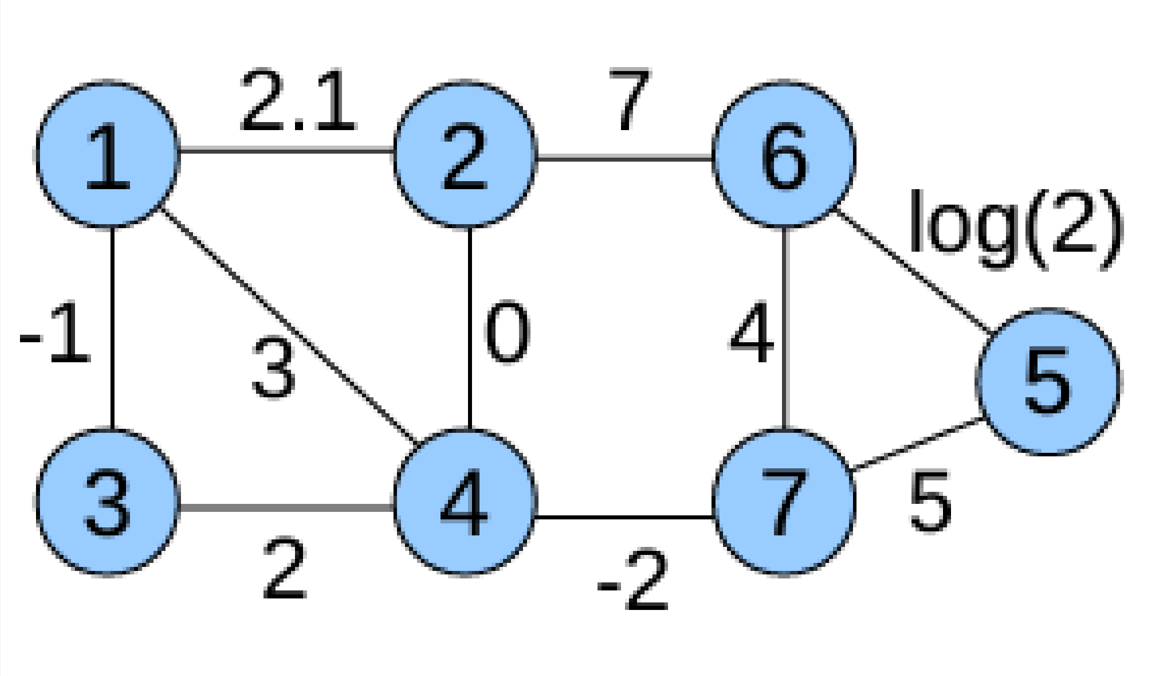
Edsger Wybe Dijkstra

11/5/1930 – 6/8/2002

- Renomado professor e pesquisador em Computação;
- Recebeu Prêmio Turing em 1972;
- Contribuições fundamentais em ling. de programação e verificação formal;
- Algoritmo de Dijkstra utilizado em vários sistemas (redes, GPS, etc);
  - Documentário: Discipline in Thought (2000) http://www.cs.utexas.edu/users/EWD/video-audio/Noo rderlichtVideo.html

Grafos com Pesos

Anotar arestas do grafo com "intensidade" do relacionamento peso da aresta (weight) função w(e) retorna peso da aresta:

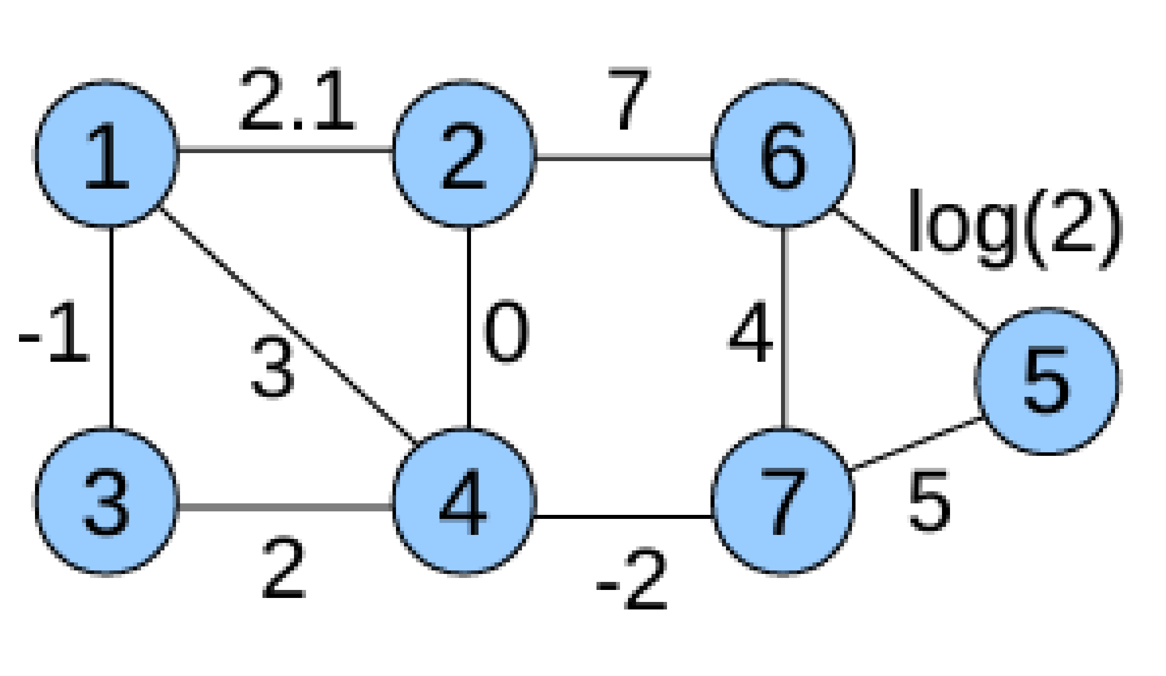


- w(2,6) = 7
- w(5,6) = log(2)
- w(2,4) = 0
- w(3,1) = -1

Comprimento com Pesos

- Comprimento de um caminho soma dos pesos das arestas que definem caminho;
- Caminho p = (v1 , v2 , ..., vk);

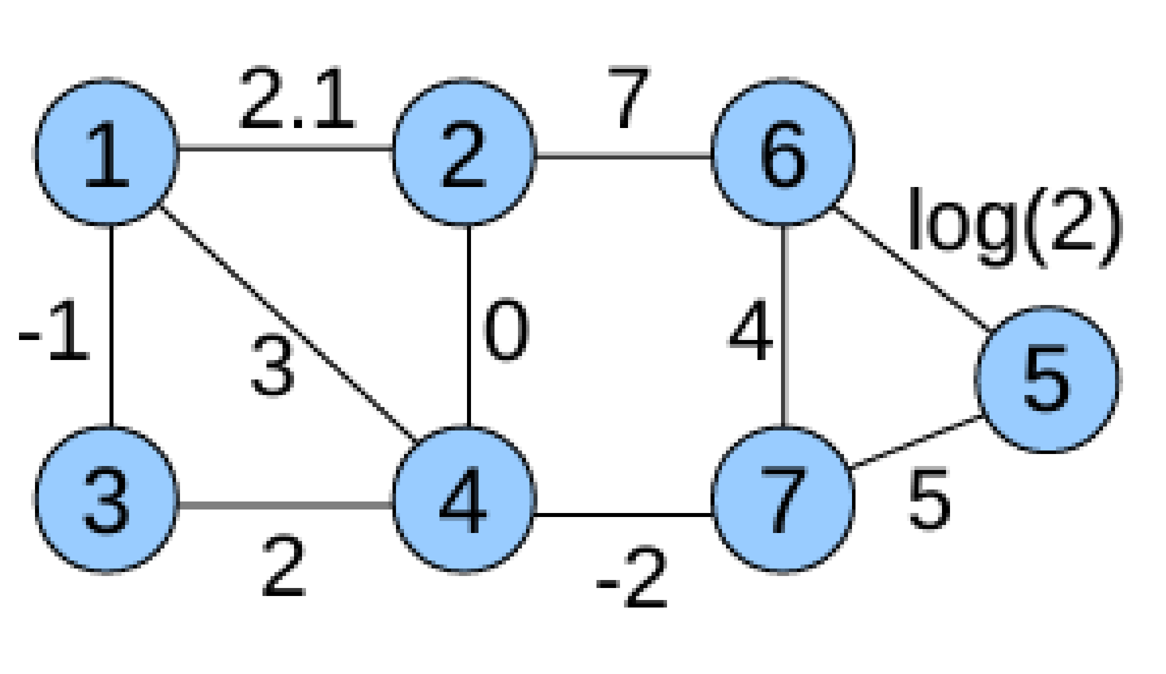
$$C(p) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$$



Comprimento com Pesos

- Comprimento de um caminho soma dos pesos das arestas que definem caminho;
- Caminho p = (v1 , v2 , ..., vk);

$$C(p) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$$

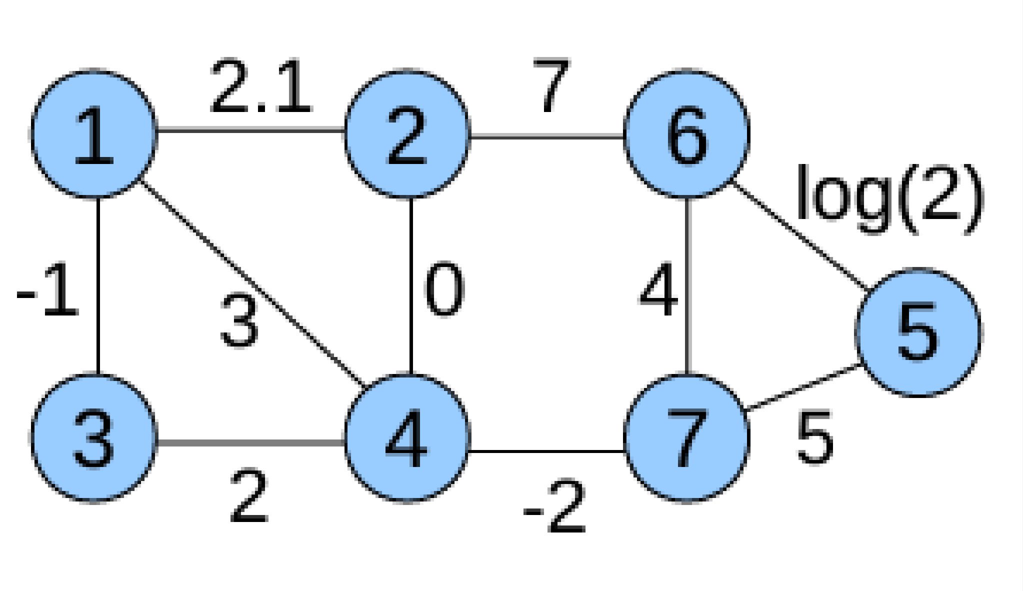


- p = (1,2,6,7) → C(p) = 7.1
- p = (1,3,4,7) → C(p) = -1
- p = (2,4,7,6,5) → C(p) = 2 + log(2)

Distância com Pesos

- Comprimento do menor caminho simples entre dois vértices;
- P(u,v) : conjunto com todos os caminhos simples entre u e v

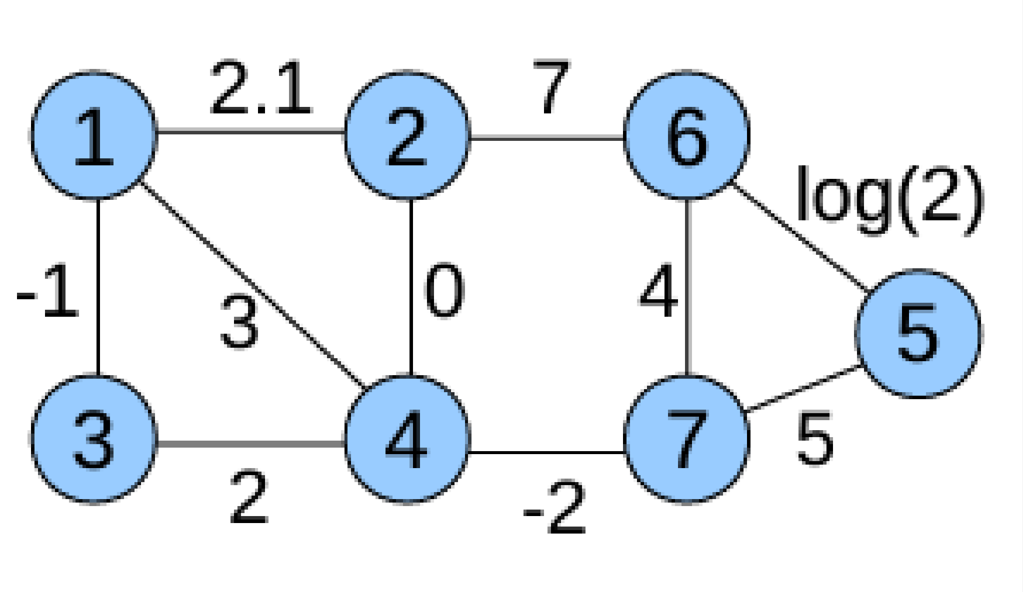
$$d(u,v) = min_{p \in P(u,v)} C(p)$$



Distância com Pesos

- Comprimento do menor caminho simples entre dois vértices;
- P(u,v) : conjunto com todos os caminhos simples entre u e v

$$d(u,v) = min_{p \in P(u,v)} C(p)$$



- d(4,1) = 1
- d(1,7) = -1
- d(3,5) = 2.1 + log(2)
- d(5,7) = log(2) - 1

- Menor caminho não necessariamente é o mais curto em arestas!

Grafos Direcionados com Pesos

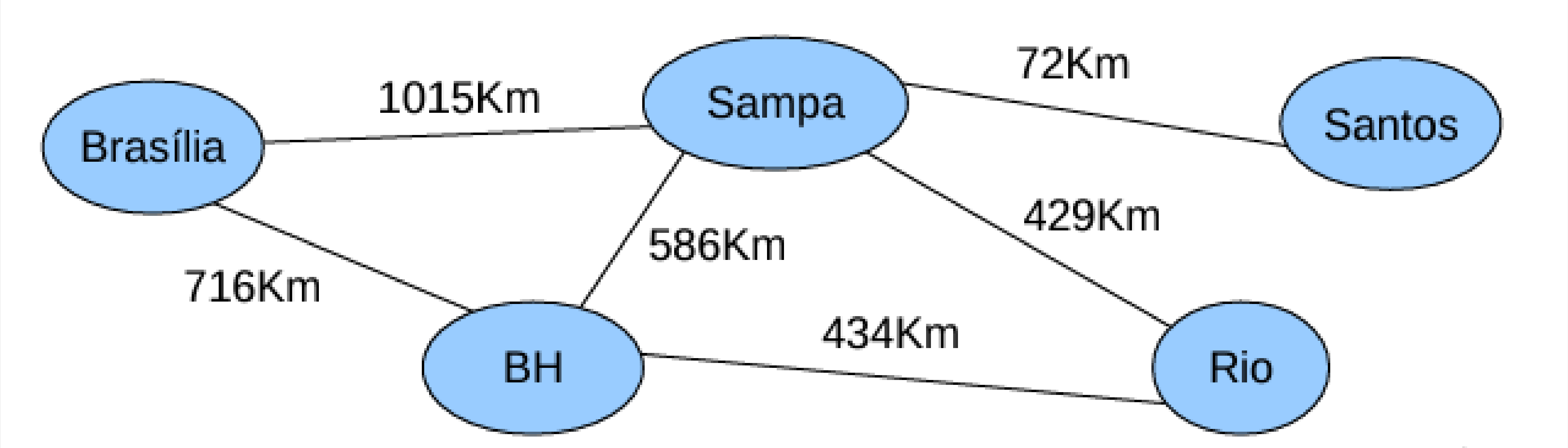
- Relacionamentos assimétricos com pesos (diferentes intensidades) :
  - ruas em uma cidade;
  - similaridade entre seguidores do twitter;
- Mesma ideia: arestas possuem "intensidade" do relacionamento
  - função w(e) retorna peso da aresta;
  - aresta direcionada, pesos potencialmente diferentes nas duas direções;

Problema - Viagem entre Cidades

- Problema 1: Como saber se duas cidades estão “conectadas” por estradas?
- Problema 2: Qual é o menor (melhor) caminho entre duas cidades?

Problema - Viagem entre Cidades

- Abstração via grafos com pesos



- Problema 1: Como saber se duas cidades estão “conectadas” por estradas? (RESOLVIDO)
- Problema 2: Qual é o menor (melhor) caminho entre duas cidades?

Distância em Grafos com Pesos

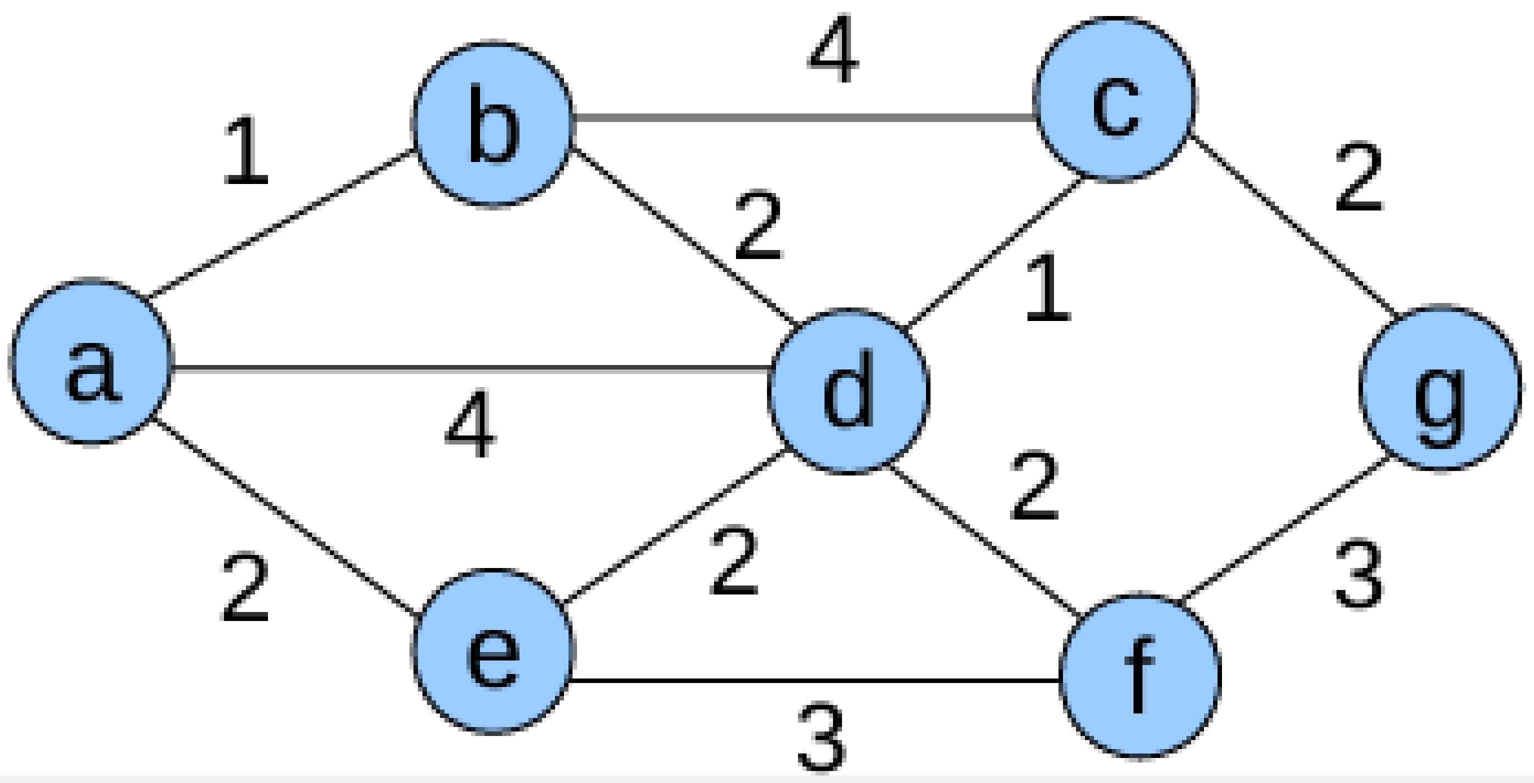
- Calcular caminho mais curto entre cidades é calcular a distância em grafos com peso
    - assumir pesos positivos;
- Dado G, com pesos;
- Determinar a distância do vértice s ao d;
- Como resolver este problema?
- Como resolvemos problemas sem pesos?

Distância em Grafos com Pesos

- Ideia: partindo de s, expandir os caminhos, incluindo vértices;
- Mas em que ordem?
    - Na ordem que garanta que iremos passar por caminhos mínimos!
- Expandir caminhos mínimos até chegar em d;

Distância em Grafos com Pesos

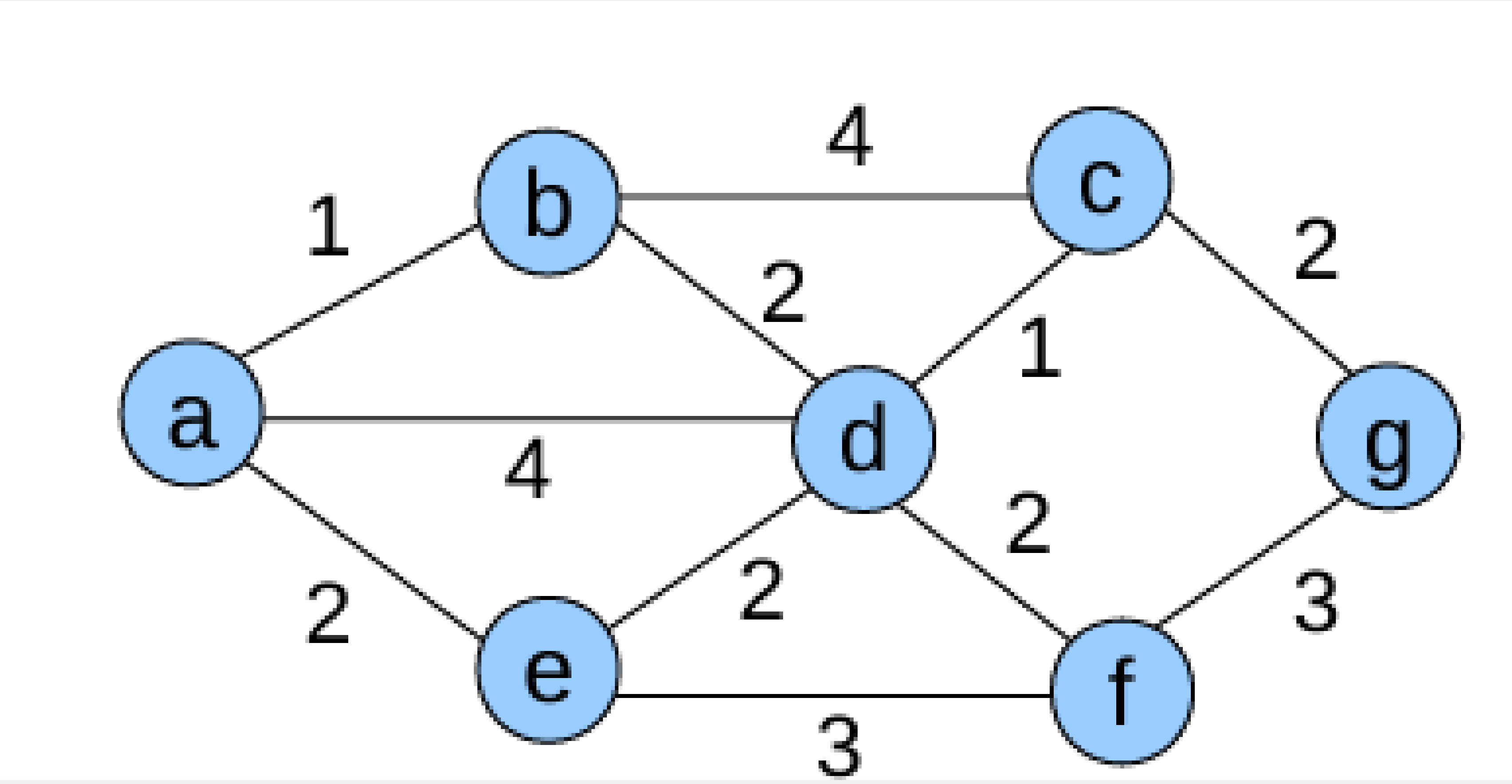
- Exemplo: distância entre a e g?



- Começar em a, expandir
- Qual próximo vértice?
- Qual vértice nos dá caminho mínimo?

Exercício

- Quais os caminhos entre a e os demais vértices?
- Qual a distância entre a e os demais vértices?



Algoritmo de Dijkstra

- Tornando a ideia em algoritmo
    - a cada passo, adicionar o vértice para o qual temos o menor caminho
- Ideias:
    - Manter dois conjuntos de vértices (descobertos, explorados)
    - Manter comprimento do menor caminho conhecido até o momento para cada vértice descoberto
    - Adicionar o vértice de menor caminho ao conjunto explorado
    - Atualizar distâncias através deste vértice, descobrindo novos vértices

Algoritmo de Dijkstra

- Condições Necessárias:
  - Não admite pesos negativos;
  - Peso Nulo (0) é possível;
  - Pode ser aplicado tanto em grafos orientados ou não orientados;

Algoritmo de Dijkstra

```
Dijkstra(G, s)
  Para cada vértice v
    dist[v] = infinito
  Define conjunto S = 0 // inicia vazio
  dist[s] = 0
  Enquanto S != V
     Selecione u em V-S, tal que dist[u] é mínima
     Adicione u em S
     Para cada vizinho v de u faça
       Se dist[v] > dist[u] + w(u,v) então
          dist[v] = dist[u] + w(u,v)
Retorna dist[]
```

- S é o conjunto dos vértices explorados;
- V é o conjunto dos vértices do grafo;
- w(u,v) é o peso da aresta (u,v);
- dist[v] é a melhor estimativa da distância de s a v;
- se v é explorado, então dist[v] é a distância de s a v;

Executando o Algoritmo

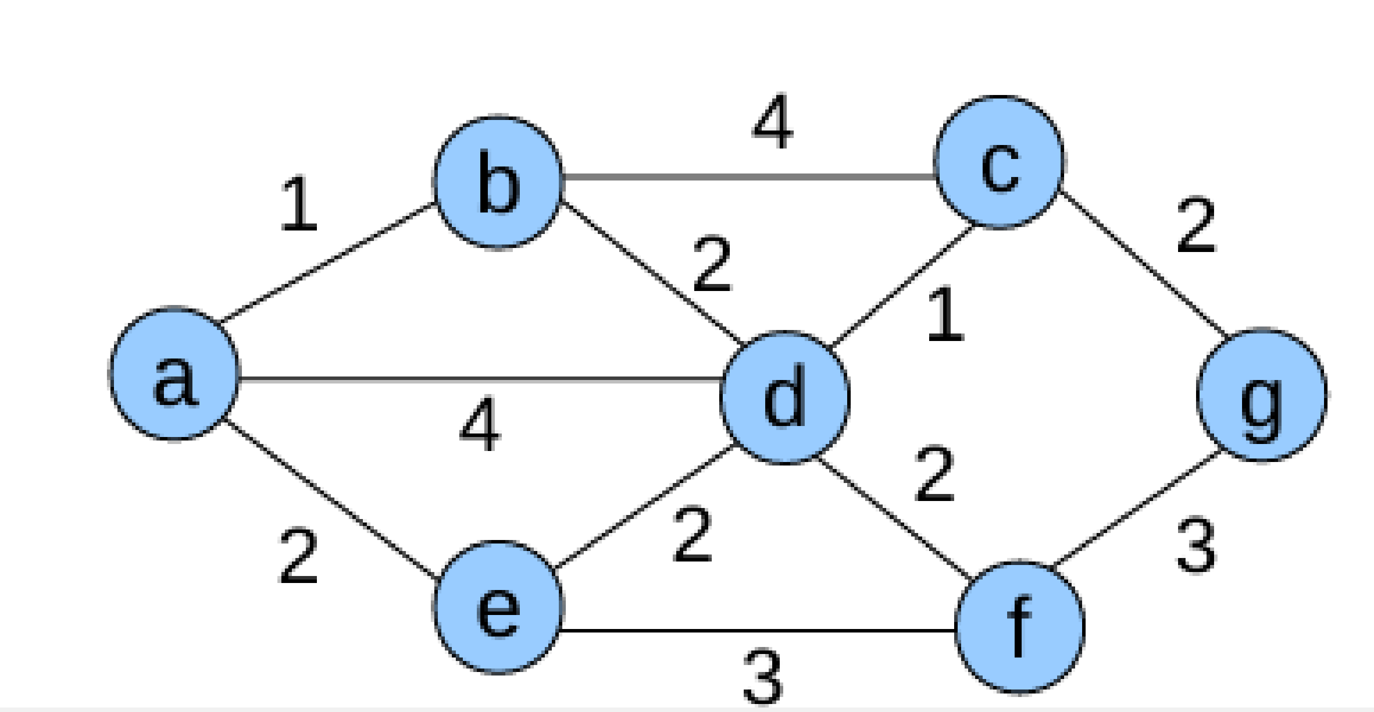


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf

Executando o Algoritmo

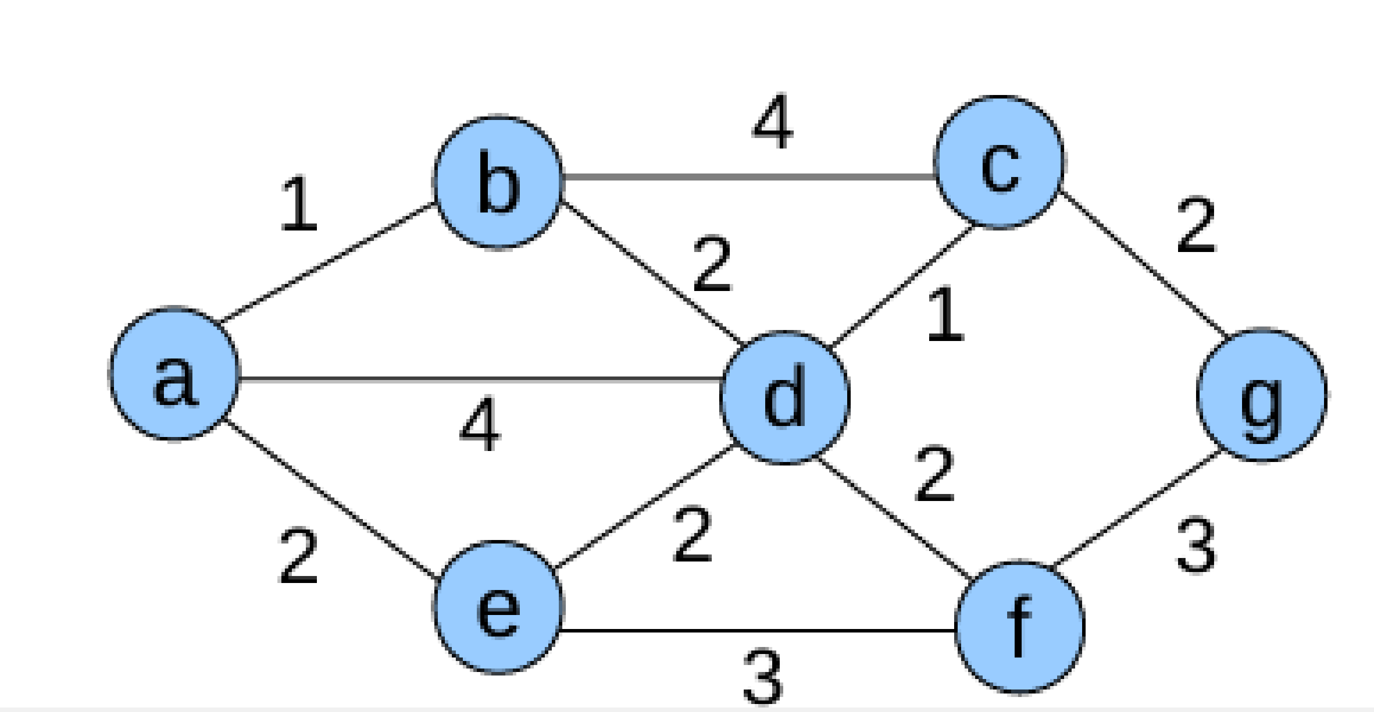


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf

Executando o Algoritmo

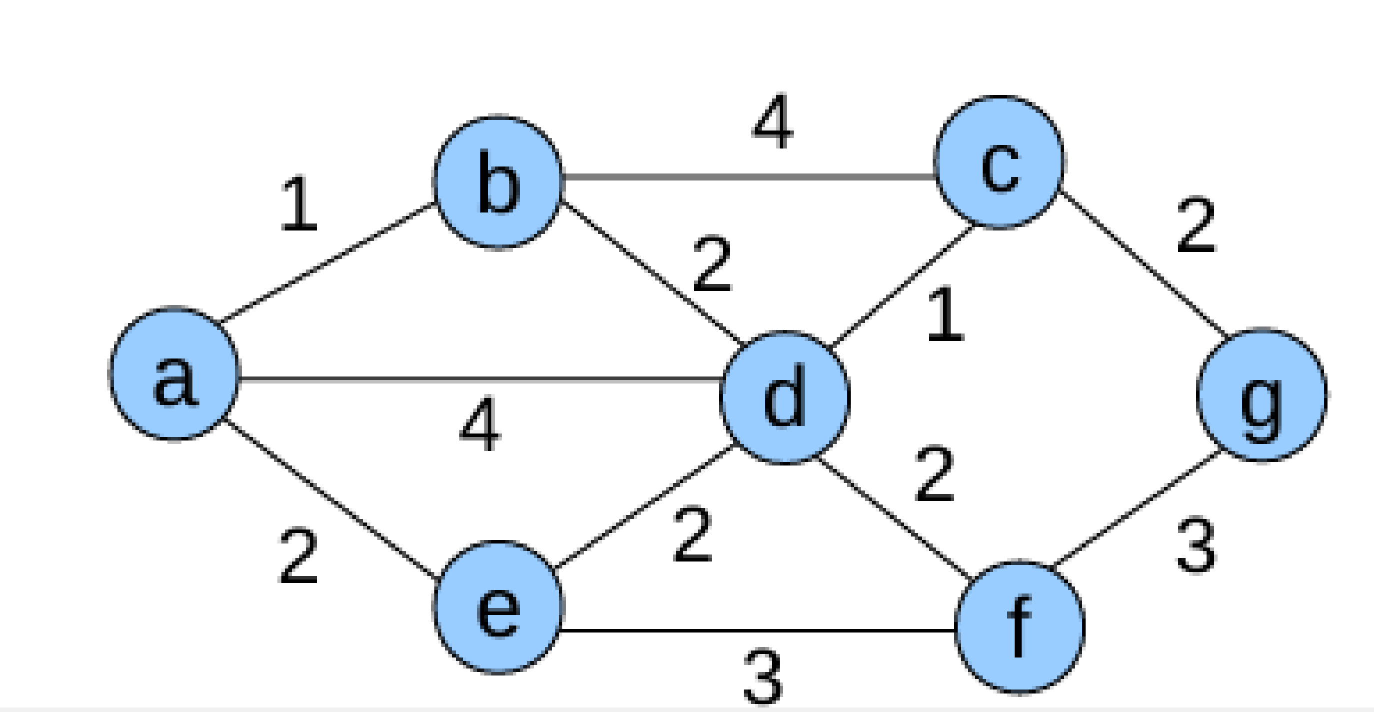


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf

Executando o Algoritmo

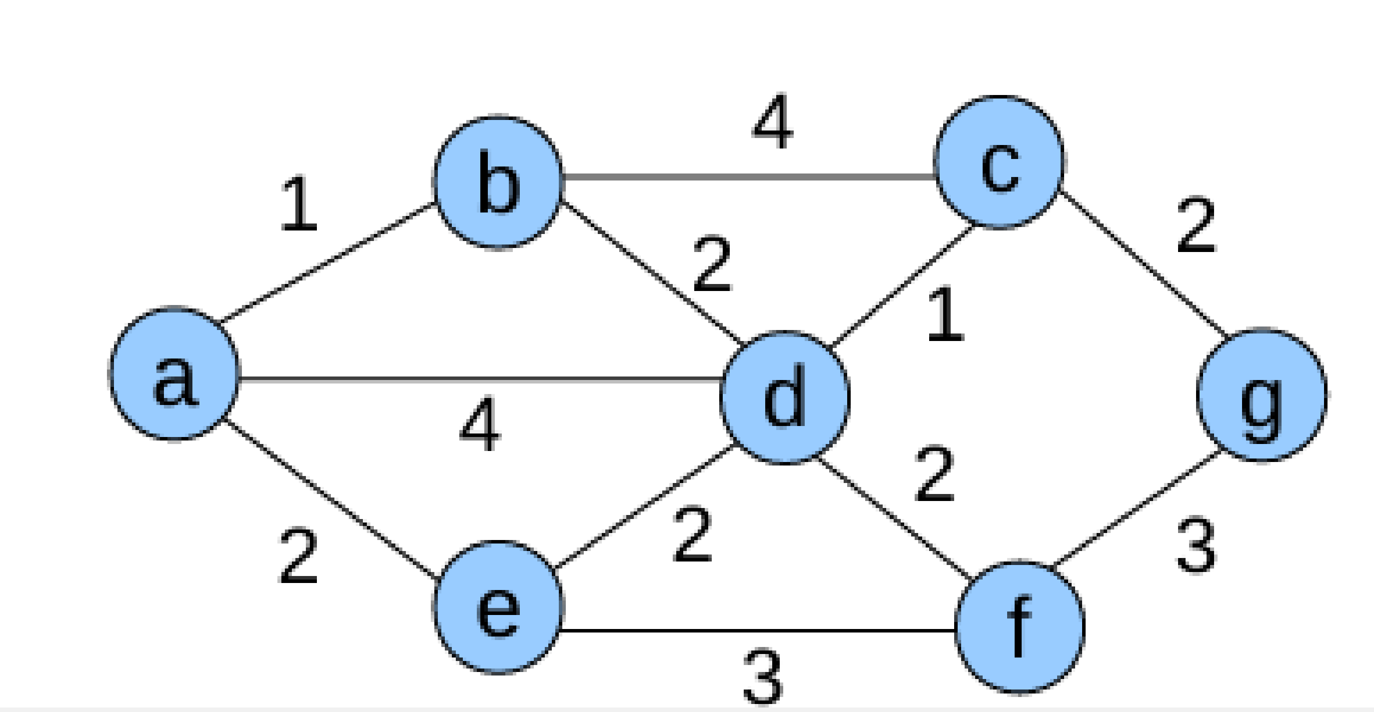


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf

Executando o Algoritmo

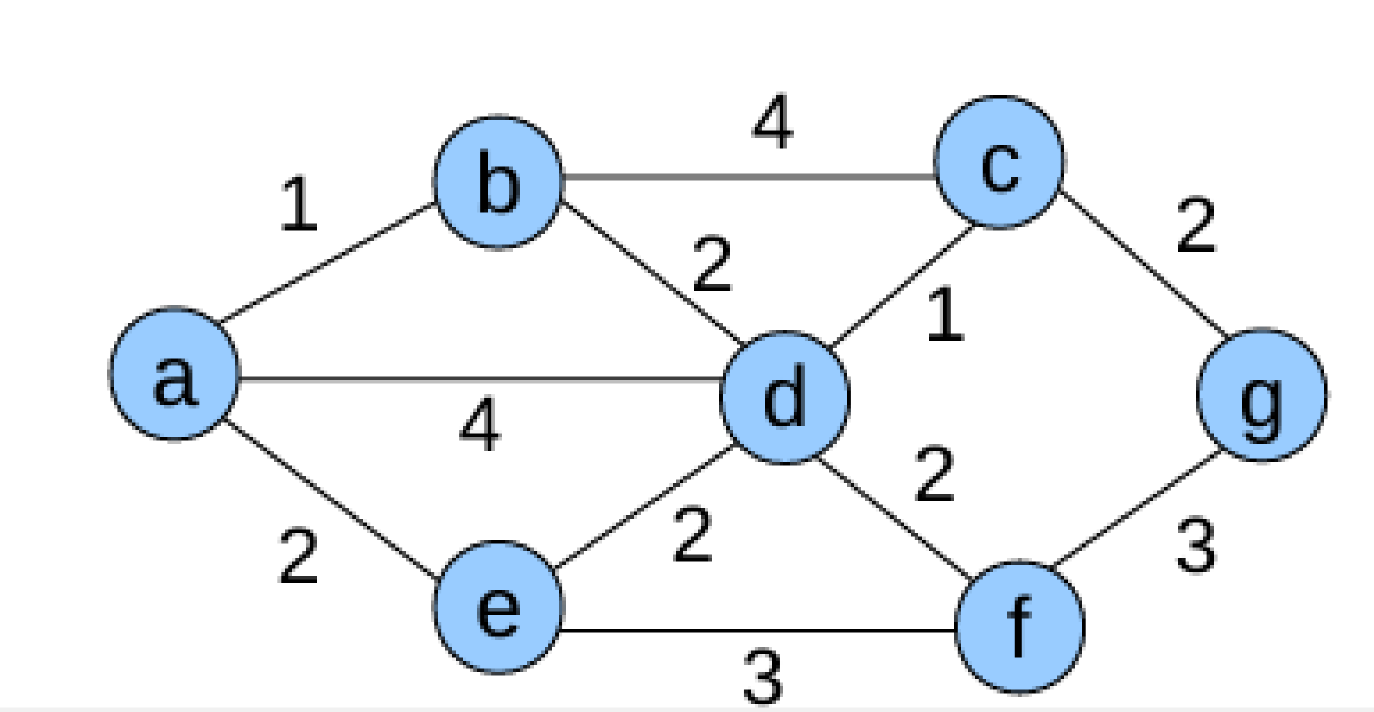


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf
4	{a,b,e,d}			4	-		5	inf

Executando o Algoritmo

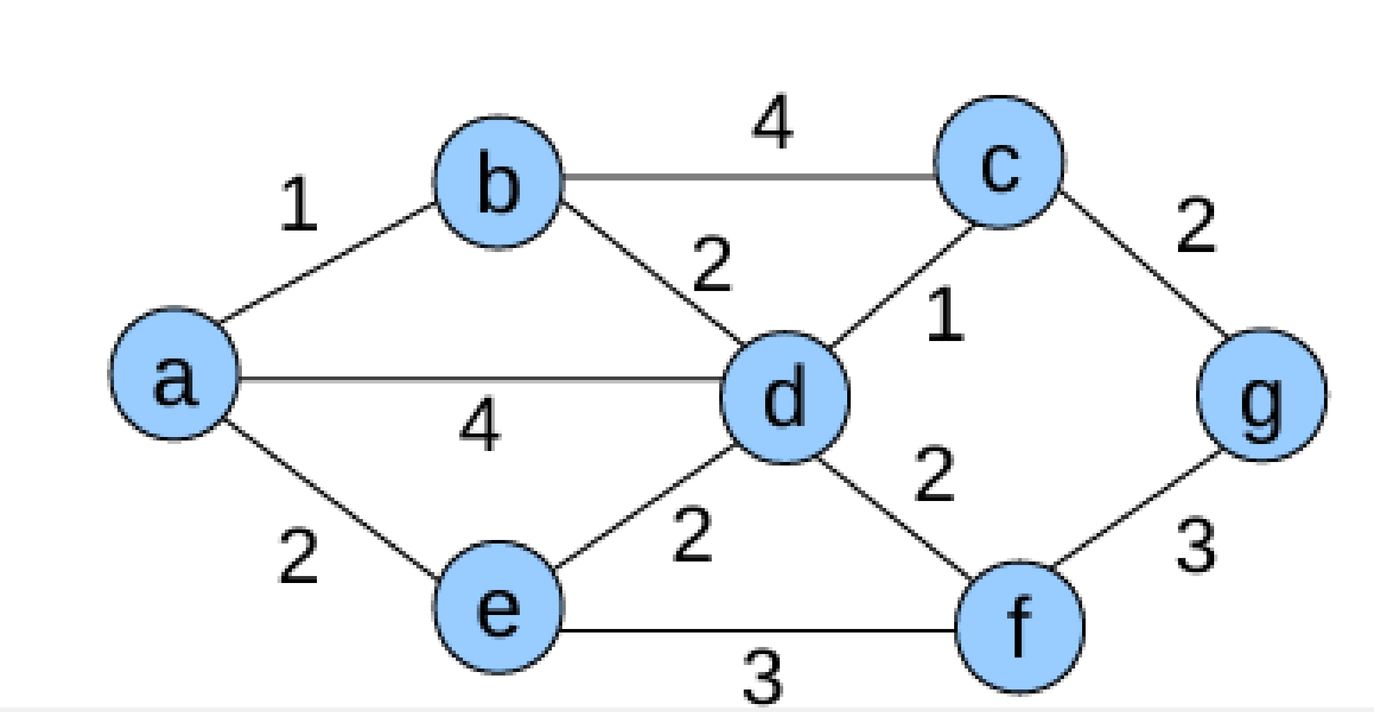


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf
4	{a,b,e,d}			4	-		5	inf
5	{a,b,e,d,c}			-			5	6

Executando o Algoritmo

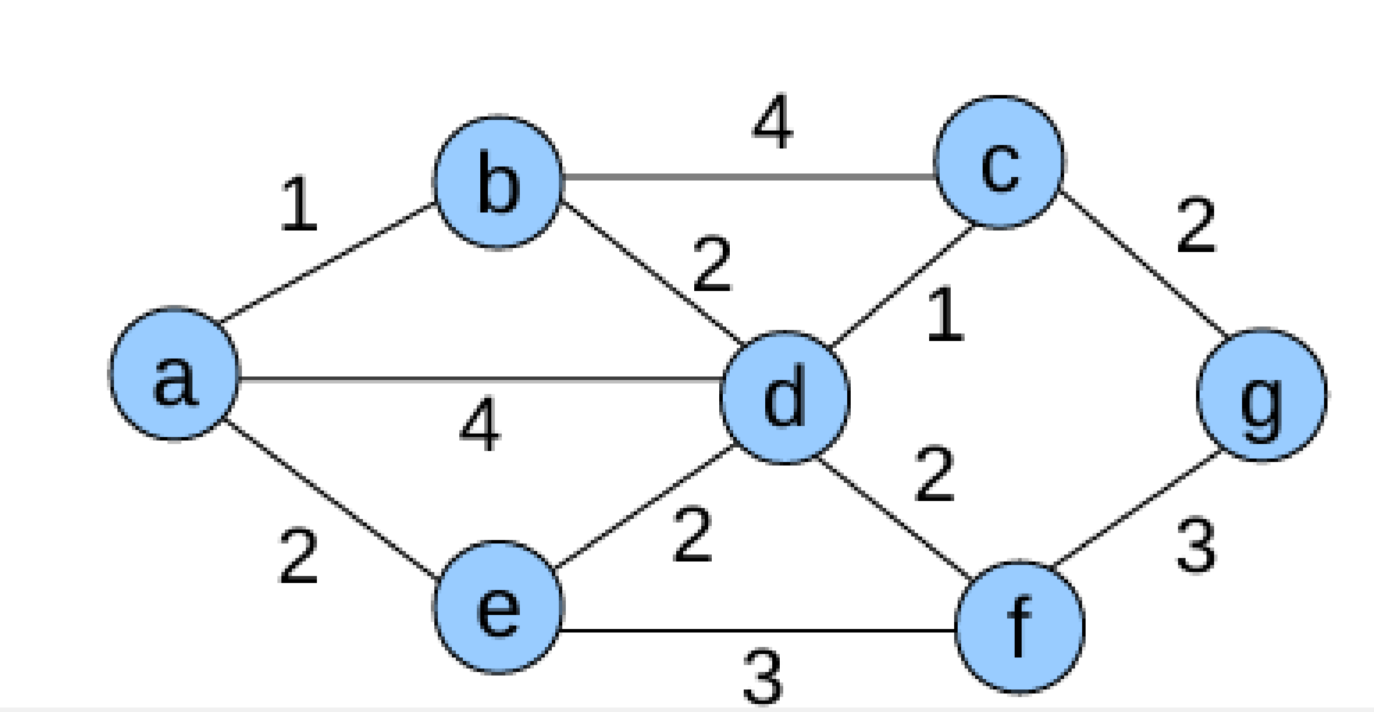


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf
4	{a,b,e,d}			4	-		5	inf
5	{a,b,e,d,c}			-			5	6
6	{a,b,e,d,c,f}						-	6

Executando o Algoritmo

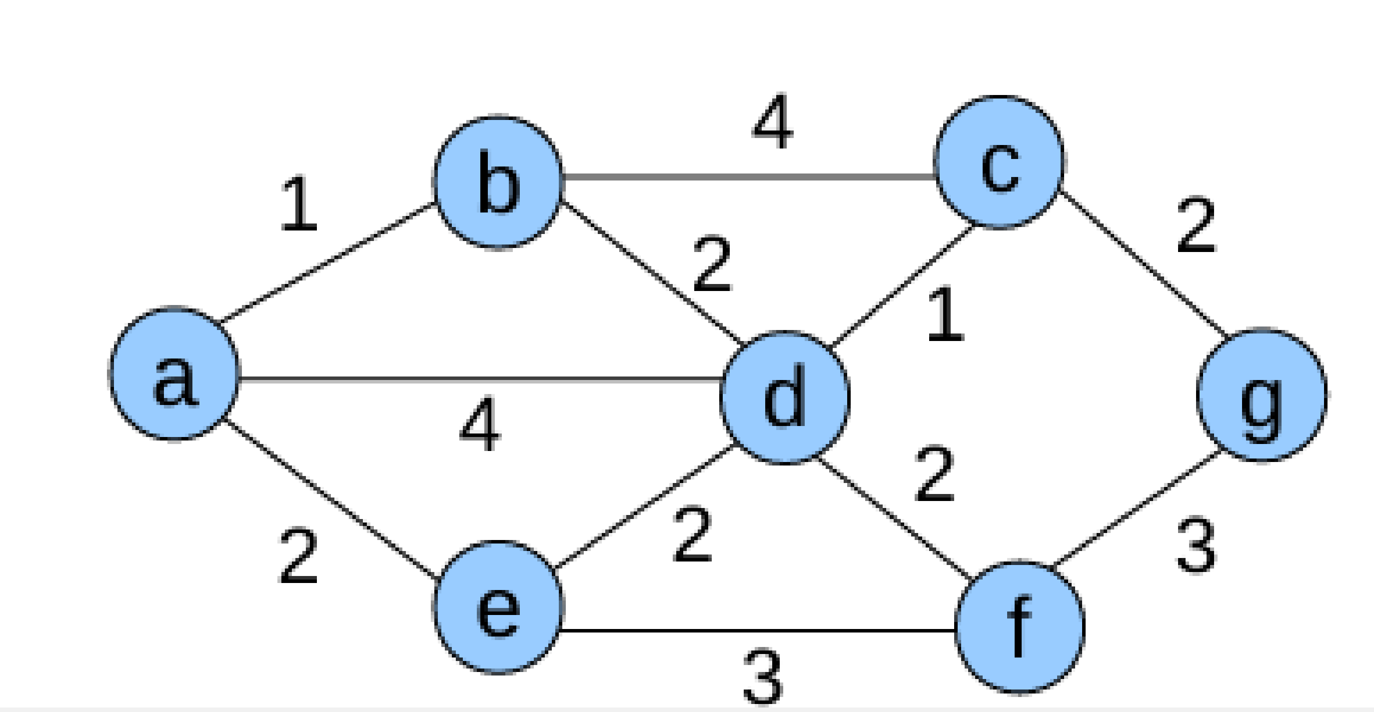
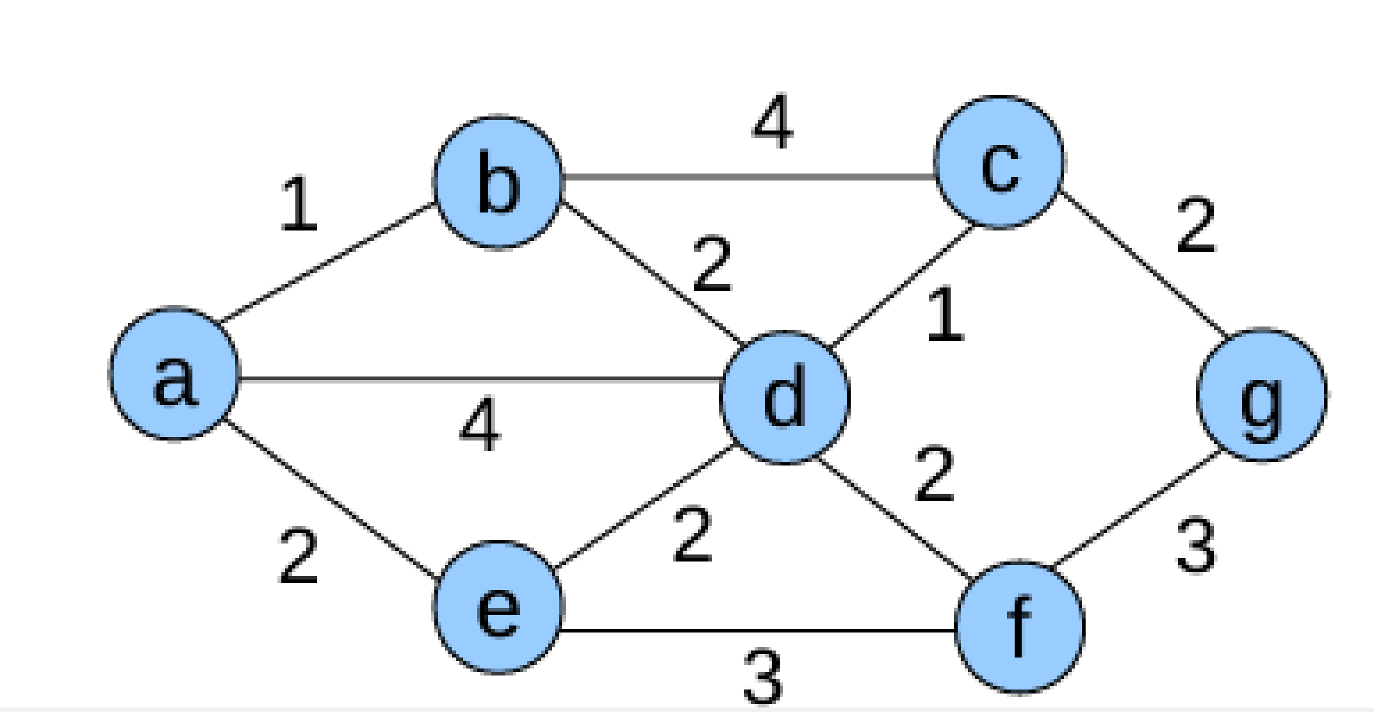


Tabela indica passos do algoritmo, atualização do vetor de distâncias e do conjunto S

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf
4	{a,b,e,d}			4	-		5	inf
5	{a,b,e,d,c}			-			5	6
6	{a,b,e,d,c,f}						-	6
7	{a,b,e,d,c,f,g}							-

E qual é o caminho???



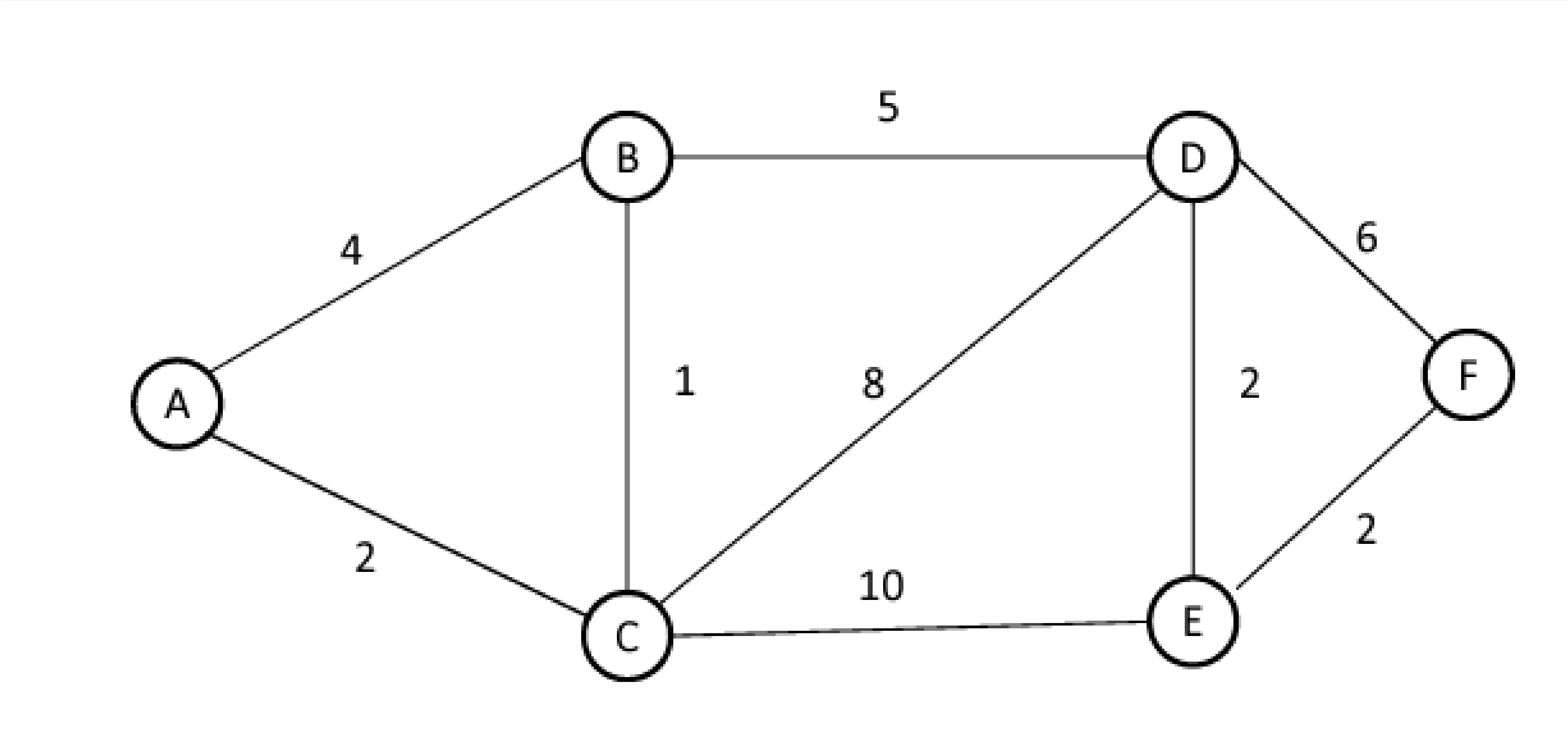
Temos que guardar a informação do predecessor no mesmo momento que atualizamos a distância!!!

vetor dist[ ] é d[ ] na tabela abaixo

Passo	Conjunto S	d[a]	d[b]	d[c]	d[d]	d[e]	d[f]	d[g]
0	{}	0	inf	inf	inf	inf	inf	inf
1	{a}	-	1	inf	4	2	inf	inf
2	{a,b}		-	5	3	2	inf	inf
3	{a,b,e}			5	3	-	5	inf
4	{a,b,e,d}			4	-		5	inf
5	{a,b,e,d,c}			-			5	6
6	{a,b,e,d,c,f}						-	6
7	{a,b,e,d,c,f,g}							-

Exercício

- Encontre o menor caminho utilizando Dijkstra entre a e f, Faça o passo a passo.



Dúvidas??