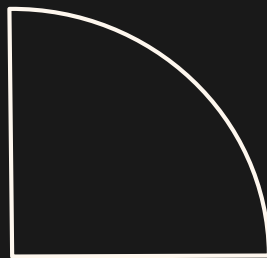
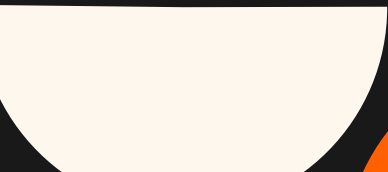




TEORIA DOS GRAFOS

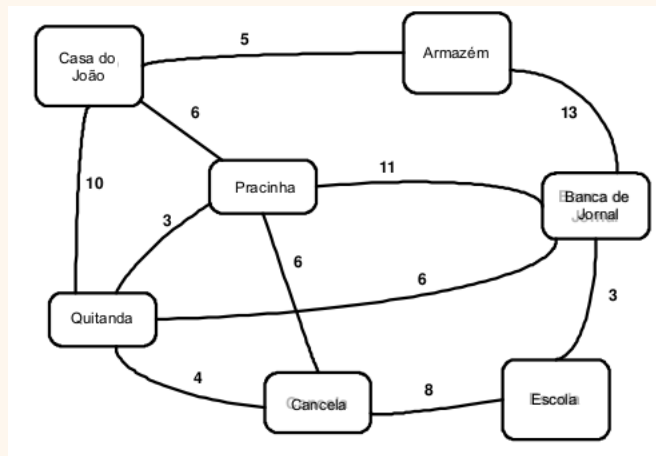
Prof^a Laura Pacifico

2025 | AGOSTO



O problema do menor caminho

- “Qual caminho oferece uma trajetória de menor comprimento?”
- Um dos algoritmos mais tradicionais para resolver esse problema foi criado por Dijkstra (Holanda), em 1952
- No problema do menor caminho, as arestas são valoradas
 - Custo, distância



O algoritmo de Dijkstra

O algoritmo de Dijkstra aqui descrito identifica o menor caminho entre dois vértices de um grafo.

Se desejamos calcular o menor caminho de a para z em um grafo conexo simples com pesos, primeiro encontramos um menor caminho entre a e um primeiro vértice, depois entre a e um segundo vértice, esse procedimento é repetido até que seja encontrado um menor caminho entre a e z .

O algoritmo de Dijkstra

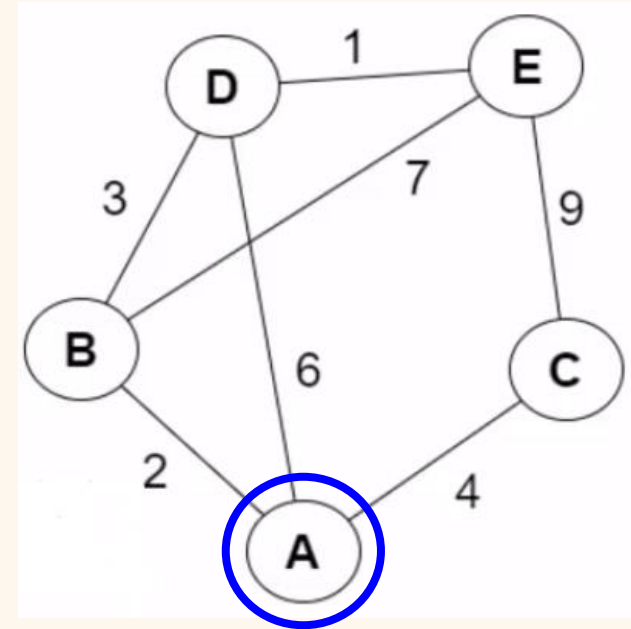
Um conjunto S de vértices é construído inserindo-se um vértice a a cada iteração.

A cada iteração também é adotado um procedimento para rotular vértices: um vértice w é rotulado com o tamanho do menor caminho de a até ele, e que contem somente vértices do conjunto S .

O vértice a a ser inserido em S é aquele com o menor rótulo.

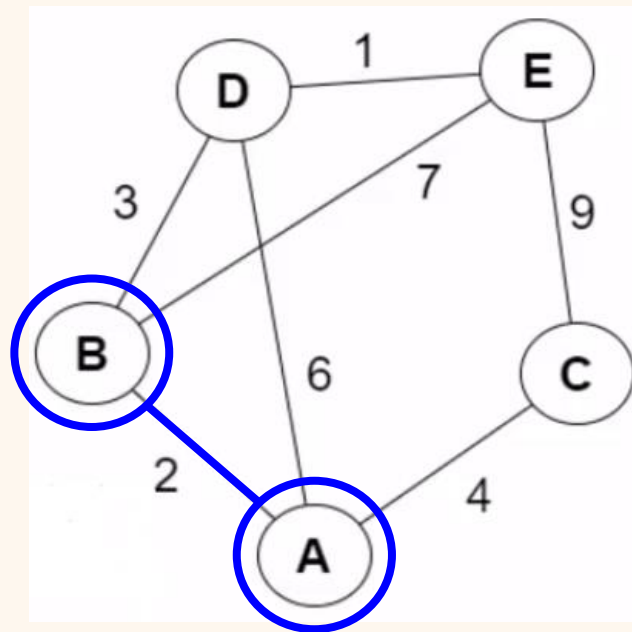
Algoritmo de Dijkstra

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
A	(0, A)				
B	(2, A)				
C	(4, A)				
D	(6, A)				
E	-				



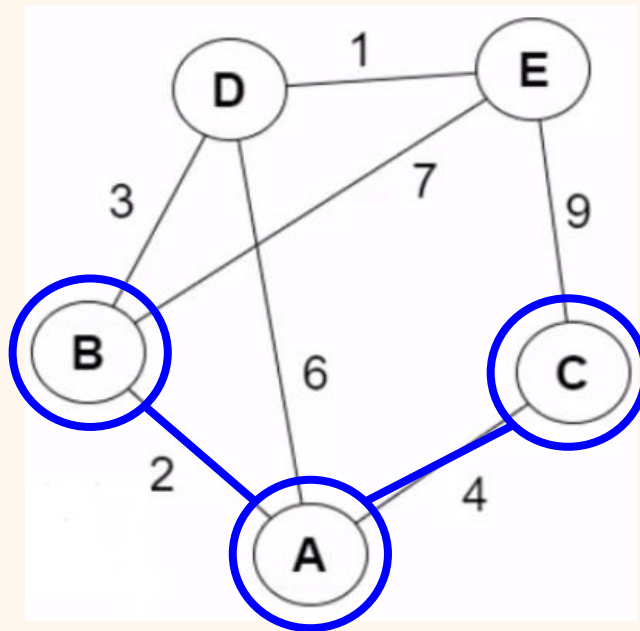
Algoritmo de Dijkstra

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
A	(0, A)	*	*	*	*
B	(2, A)	(2, A)	*	*	*
C	(4, A)	(4, A)			
D	(6, A)	(5, B)			
E	-	(9, B)			



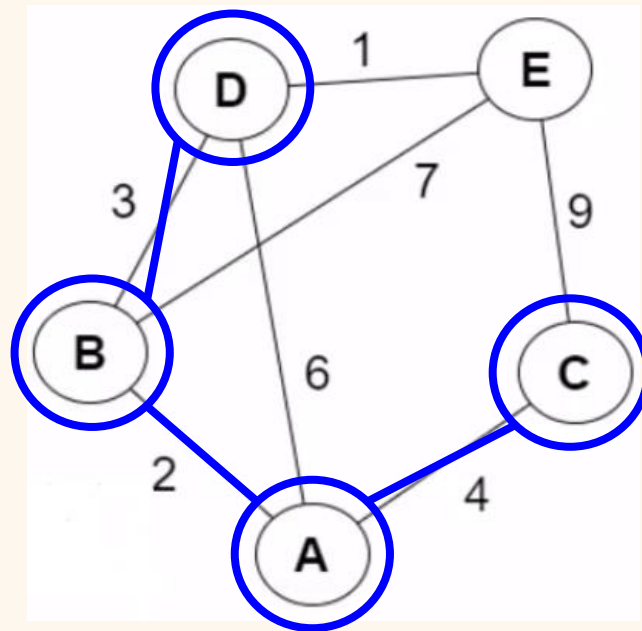
Algoritmo de Dijkstra

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
A	(0, A)	*	*	*	*
B	(2, A)	(2, A)	*	*	*
C	(4, A)	(4, A)	(4, A)	*	*
D	(6, A)	(5, B)	(5, B)		
E	-	(9, B)	(9, B)		



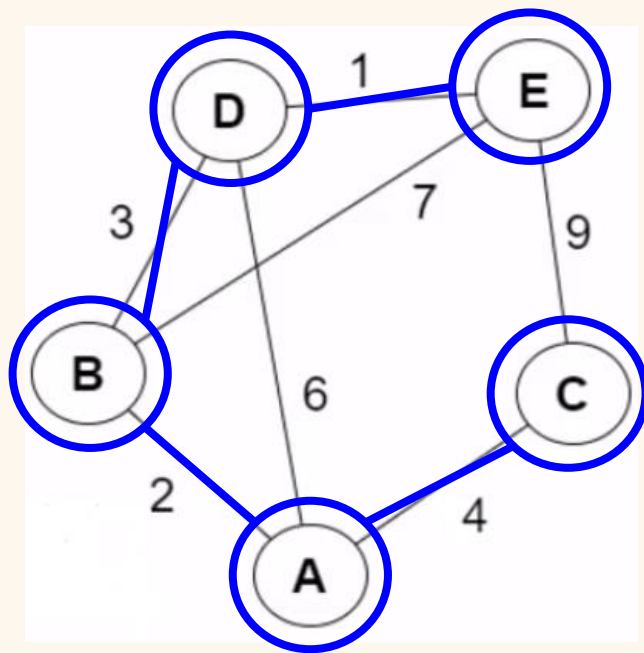
Algoritmo de Dijkstra

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
A	(0, A)	*	*	*	*
B	(2, A)	(2, A)	*	*	*
C	(4, A)	(4, A)	(4, A)	*	*
D	(6, A)	(5, B)	(5, B)	(5, B)	*
E	-	(9, B)	(9, B)	(6, D)	



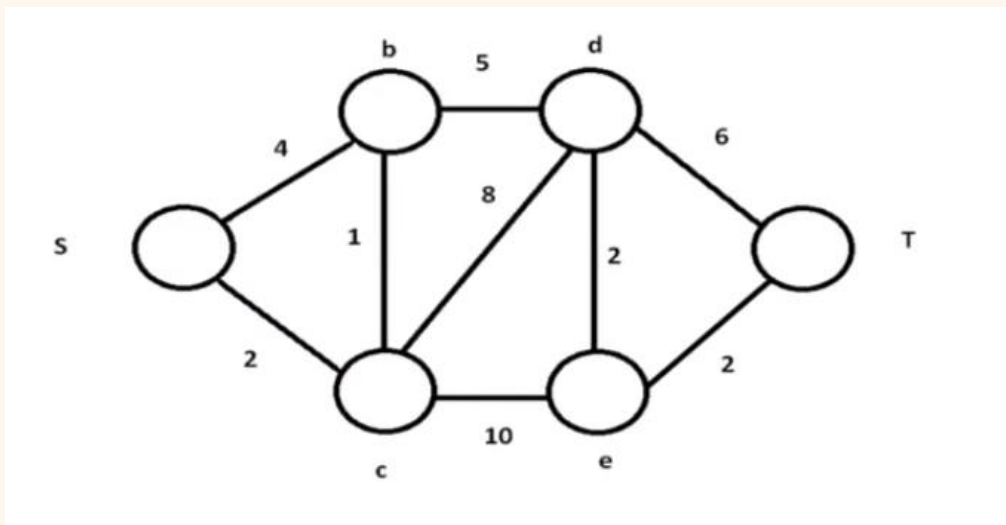
Algoritmo de Dijkstra

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
A	(0, A)	*	*	*	*
B	(2, A)	(2, A)	*	*	*
C	(4, A)	(4, A)	(4, A)	*	*
D	(6, A)	(5, B)	(5, B)	(5, B)	*
E	-	(9, B)	(9, B)	(6, D)	(6, D)



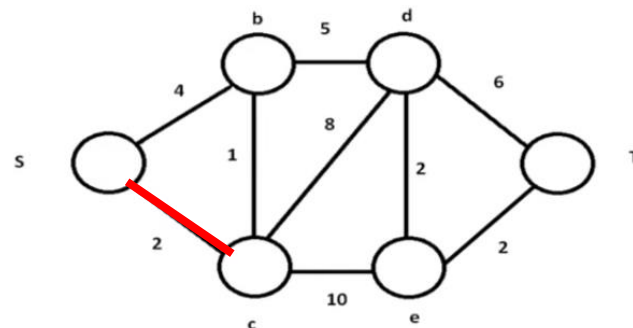
Atividade

- Utilizando o algoritmo de Dijkstra, calcule a menor distância entre os vértices S e T



Atividade

	1	2	3	4	5	6
S	(0, S)	*	*	*	*	*
B	(4, S)	(3, C)	(3, C)	*	*	*
C	(2, S)	(2, S)	*	*	*	*
D	∞	(10, C)	(8, B)	(8, B)	*	*
E	∞	(12, C)	(12, C)	(10, D)	(10, D)	*
T	∞	∞	∞	(14, D)	(12, E)	(12, E)



Algoritmo de Dijkstra

- Para cada vértice v do grafo, mantemos um atributo $d[v]$ que é um **limite superior** para o peso do caminho mais curto do nó inicial s a v
 - Dizemos que $d[v]$ é uma **estimativa do caminho mais curto**, inicialmente feito ∞
- Também armazenamos o vértice que precede v ($p[v]$ - **precedente** de v) no caminho mais curto de s a v

Algoritmo de Dijkstra

- Faça a estimativa de distância de s a qualquer vértice ser infinita
 - Exceto, claro, a distância de s a s , que é 0
 - Ou seja, $d[s] = 0$ e $d[v] = \infty$ para todo $v \neq s$
- Faça os precedentes dos nós serem um valor qualquer
 - Na prática, podemos fazer $p[v] = -1$, já que não temos vértice de índice -1 na estrutura grafo
- Marque todos os vértices como “abertos”

Algoritmo de Dijkstra

- Enquanto houver vértice aberto:
 - Escolha o vértice aberto u cuja estimativa seja a menor dentre os demais abertos
 - Feche u
 - Para todo nó aberto v na adjacência de u :
 - Some $d[u]$ ao peso da aresta (u, v)
 - Caso a soma seja menor que $d[v]$, atualize $d[v]$ e faça $p[v] = u$
 - Procedimento chamado de **relaxamento** da aresta (u, v)

Algoritmo de Dijkstra - Resumo

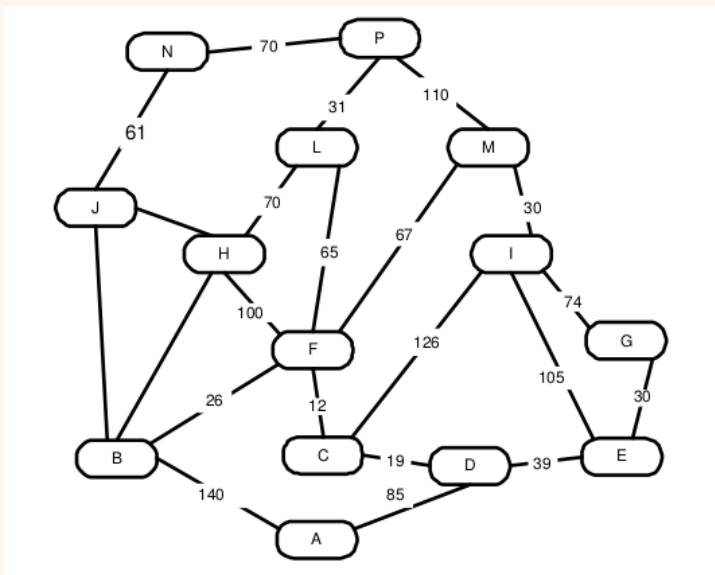
- Inicialize o grafo com $d[s] = 0$, $d[v] = \infty$, para todo $v \neq s$, e $p[v] = -1$ para todo v
- Faça $aberto[v] = true$ para todo v no grafo
- Enquanto houver vértice aberto:
 - Escolha u cuja estimativa seja a menor dentre os abertos
 - Feche u
 - Para todo nó aberto v na adjacência de u :
 - Relaxe a aresta (u, v)

Algoritmo de Dijkstra - Observações

- O algoritmo de Dijkstra tem uma limitação importante
 - Os pesos das arestas devem ser não-negativos
- A implementação das funções auxiliares influencia diretamente na eficiência do algoritmo
 - Implementação com **filas de prioridade** normalmente alcança resultados mais rápidos
- O algoritmo encontra o menor caminho de um dado vértice a todos os outros nós
 - Ele não encontra o menor caminho entre dois vértices quaisquer

Atividade

- Use o algoritmo de Dijkstra para descobrir qual o menor caminho do vértice A a todos os outros vértices



Dúvidas?



Laura Alves Pacifico
laps@cesar.school
Slack: Laura Pacifico