

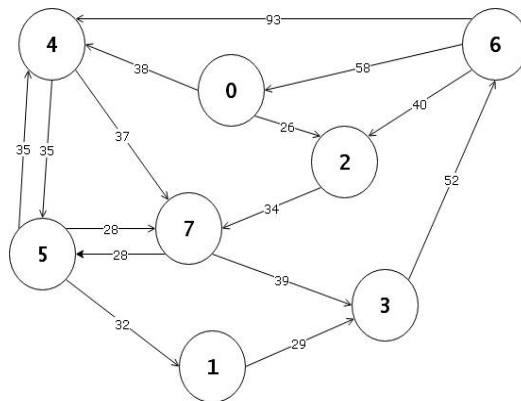
## Roteiro para Laboratório: Caminho de Custo Mínimo em Grafo Ponderado Direcionado

### Parte 1: Problema do caminho mínimo sob custos positivos:

Dados vértices  $s$  e  $t$  de um grafo com custos **positivos** nos arcos, encontrar um caminho mínimo de  $s$  a  $t$ . Se não houver caminho algum de  $s$  a  $t$ , o problema não tem solução.

**Exemplo:** A tabela a seguir mostra um grafo com custos positivos nos arcos.

4→5	5→4	4→7	5→7	7→5	5→1	0→4	0→2	7→3	1→3	2→7	6→2	3→6	6→0	6→4
35	35	37	28	28	32	38	26	39	29	34	40	52	58	93



Considere os caminhos que vão de 0 a 6. O caminho 0-2-7-5-1-3-6 não tem custo mínimo pois o caminho 0-2-7-3-6 é mais barato. O custo desse último caminho é  $26+34+39+52 = 151$ . Esse caminho é mínimo?

### Algoritmo de Dijkstra para Cálculo do Caminho Mínimo:

O algoritmo de Dijkstra identifica, a partir de um vértice de origem do grafo, qual é o custo mínimo entre esse vértice e todos os outros do grafo. No início, o conjunto  $S$  contém somente esse vértice, chamado *origem*. A cada passo, seleciona-se o mais próximo da origem entre os vértices restantes, e este vértice é inserido no conjunto  $S$ . Em seguida, atualiza-se para cada vértice restante, sua distância em relação à origem. Se passando pelo novo vértice acrescentado, a distância fica menor, é essa nova distância que será registrada.

#### O Algoritmo de Dijkstra

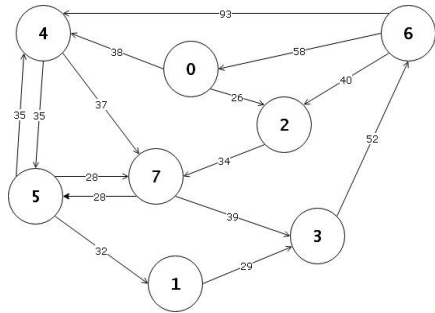
**Require:**  $G = (V, E)$ , um grafo orientado ponderado  
**Require:**  $C$ , uma matriz de custos associados aos vértices  $E$   
**Ensure:**  $D$  um vetor com as custos mínimos entre cada vértice em  $E$  e o vértice origem 1

```

1:  $S \leftarrow \{1\}$ 
2: for  $i \leftarrow 2$  to  $n$  do
3:    $D[i] \leftarrow C[1, i]$ 
4: end for
5: for  $i \leftarrow 2$  to  $n$  do
6:   encontre um vértice  $w \in V - S$  tal que  $D[w]$  é mínimo
7:    $S \leftarrow S \cup \{w\}$ 
8:   for all  $v \in V - S$  do
9:      $D[v] \leftarrow \min(D[v], D[w] + C[w, v])$ 
10:  end for
11: end for

```

**Menor custo a partir do vértice 0 até o vértice 6 - Passos:**



Vt	0	1	2	3	4	5	6	7
0			26		38			
1				29				
2								34
3							52	
4						35		37
5		32			35			28
6	58		40		93			
7				39	28			

Vt =vértices

**Estruturas de Controle e Cálculo:**

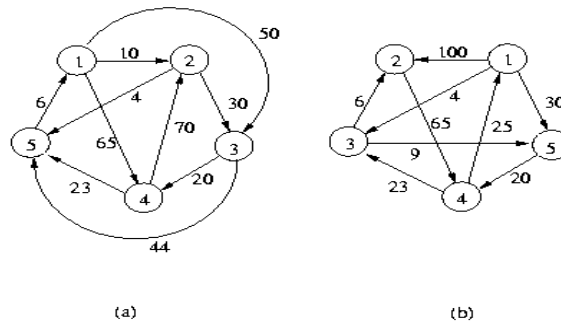
Passo	vOrigem	vAlcançados	M [0..7] visita marcada A [0..7] vértice anterior L [0..7] cálculo custo																																				
Início	0	{2,4}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>A</th> <td>-1</td> <td>-1</td> <td>0</td> <td>-1</td> <td>0</td> <td>-1</td> <td>-1</td> <td>-1</td> </tr> <tr> <th>L</th> <td></td> <td></td> <td>26</td> <td></td> <td>38</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	0	0	0	0	0	0	0	A	-1	-1	0	-1	0	-1	-1	-1	L			26		38			
vt	0	1	2	3	4	5	6	7																															
M	1	0	0	0	0	0	0	0																															
A	-1	-1	0	-1	0	-1	-1	-1																															
L			26		38																																		
1	2	{7}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>A</th> <td>-1</td> <td>-1</td> <td>0</td> <td>-1</td> <td>0</td> <td>-1</td> <td>-1</td> <td>2</td> </tr> <tr> <th>L</th> <td></td> <td></td> <td>26</td> <td></td> <td>38</td> <td></td> <td></td> <td>60</td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	0	1	0	0	0	0	0	A	-1	-1	0	-1	0	-1	-1	2	L			26		38			60
vt	0	1	2	3	4	5	6	7																															
M	1	0	1	0	0	0	0	0																															
A	-1	-1	0	-1	0	-1	-1	2																															
L			26		38			60																															
2	4	{5,7}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>A</th> <td>-1</td> <td>-1</td> <td>0</td> <td>-1</td> <td>0</td> <td>4</td> <td>-1</td> <td>2</td> </tr> <tr> <th>L</th> <td></td> <td></td> <td>26</td> <td></td> <td>38</td> <td>73</td> <td></td> <td>60</td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	0	1	0	1	0	0	0	A	-1	-1	0	-1	0	4	-1	2	L			26		38	73		60
vt	0	1	2	3	4	5	6	7																															
M	1	0	1	0	1	0	0	0																															
A	-1	-1	0	-1	0	4	-1	2																															
L			26		38	73		60																															
3	7	{3,5}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th>A</th> <td>-1</td> <td>-1</td> <td>0</td> <td>7</td> <td>0</td> <td>4</td> <td>-1</td> <td>2</td> </tr> <tr> <th>L</th> <td></td> <td></td> <td>26</td> <td>99</td> <td>38</td> <td>73</td> <td></td> <td>60</td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	0	1	0	1	0	0	1	A	-1	-1	0	7	0	4	-1	2	L			26	99	38	73		60
vt	0	1	2	3	4	5	6	7																															
M	1	0	1	0	1	0	0	1																															
A	-1	-1	0	7	0	4	-1	2																															
L			26	99	38	73		60																															
4	5	{1,4,7}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <th>A</th> <td>-1</td> <td>5</td> <td>0</td> <td>7</td> <td>0</td> <td>4</td> <td>-1</td> <td>2</td> </tr> <tr> <th>L</th> <td></td> <td>105</td> <td>26</td> <td>99</td> <td>38</td> <td>73</td> <td></td> <td>60</td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	0	1	0	1	1	0	1	A	-1	5	0	7	0	4	-1	2	L		105	26	99	38	73		60
vt	0	1	2	3	4	5	6	7																															
M	1	0	1	0	1	1	0	1																															
A	-1	5	0	7	0	4	-1	2																															
L		105	26	99	38	73		60																															
5	3	{6}	<table border="1"> <thead> <tr> <th>vt</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>M</th> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <th>A</th> <td>-1</td> <td>5</td> <td>0</td> <td>7</td> <td>0</td> <td>4</td> <td>3</td> <td>2</td> </tr> <tr> <th>L</th> <td></td> <td>105</td> <td>26</td> <td>99</td> <td>38</td> <td>73</td> <td>151</td> <td>60</td> </tr> </tbody> </table>	vt	0	1	2	3	4	5	6	7	M	1	1	1	1	1	1	0	1	A	-1	5	0	7	0	4	3	2	L		105	26	99	38	73	151	60
vt	0	1	2	3	4	5	6	7																															
M	1	1	1	1	1	1	0	1																															
A	-1	5	0	7	0	4	3	2																															
L		105	26	99	38	73	151	60																															

**Resultado:**

- 0 -> 1 : O custo mínimo é 105
- 0 -> 2: O custo mínimo é 26
- 0 -> 3: O custo mínimo é 99
- 0 -> 4: O custo mínimo é 38
- 0 -> 5: O custo mínimo é 73
- 0 -> 6: O custo mínimo é 151
- 0 -> 7: O custo mínimo é 60

### Exercícios:

1) Para cada grafo da figura abaixo, mostre o quadro dos passos da execução do algoritmo de Dijkstra para os menores caminhos a partir do vértice 1, e o resultado final.



2) Se o custo de cada arco de um *grafo ponderado direcionado* for multiplicado por uma constante estritamente positiva, as soluções do problema do caminho mínimo não se alteram. Esta afirmação é verdadeira?

### Parte 2: Estruturas de dados para o cálculo do caminho mínimo sob custos positivos:

Para o cálculo do caminho mínimo, serão utilizados 3 vetores do tamanho do número de vértices:

- (M) Vetor para armazenar se o vértice já foi visitado
- (A) Vetor para armazenar o vértice anterior utilizado para o custo mínimo atingido
- (L) Vetor para armazenar o custo mínimo alcançado em cada vértice

Trecho de código para definição, alocação de memória e inicialização dos vetores:

```
void Caminho_mais_Curto(TipoGrafo *grafo, int origem, int destino)
{
    int i;
    int *M, *L, *A;
    M = (int *)malloc(grafo->NumVertices*sizeof(int));
    L = (int *)malloc(grafo->NumVertices*sizeof(int));
    A = (int *)malloc(grafo->NumVertices*sizeof(int));

    // preenchimento preliminar dos vetores
    for (i=0; i<grafo->NumVertices; i++)
    {
        M[i] = 0;           // false; determina se um vertice ja foi visitado
        A[i] = -1;         // determina o caminho mais curto entre origem e destino
        L[i] = 300000;     // infinito determina o comprimento do caminho mais curto
    }
    // .....
}
```

### Exercício:

Complete a função `Caminho_mais_Curto` com o cálculo do caminho mínimo entre dois vértices (origem e destino) e a impressão deste caminho mínimo.