

# Estruturas de Dados

## Algoritmo de Dijkstra

Universidade Estadual Vale do Acaraú – UVA

---

Paulo Regis Menezes Sousa

paulo\_regis@uvanet.br

# Algoritmo de Dijkstra

- O *Algoritmo de Dijkstra*, foi criado pelo cientista da computação holandês Edsger Wybe Dijkstra em 1956 e publicado em 1959.

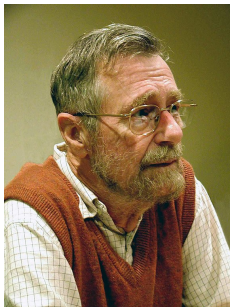


Figura 1: Edsger W. Dijkstra

O algoritmo de Dijkstra resolve o problema do caminho mínimo em um grafo dirigido ponderado  $G = (V, E)$  para o caso no qual todos os pesos de arestas são não negativos.

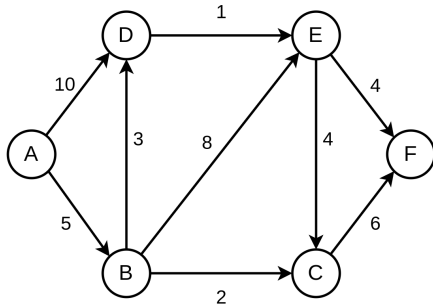
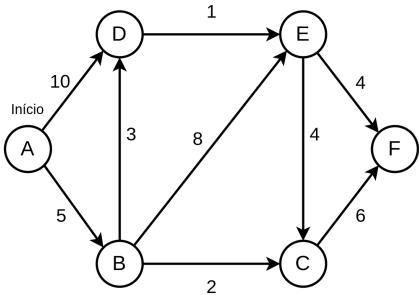
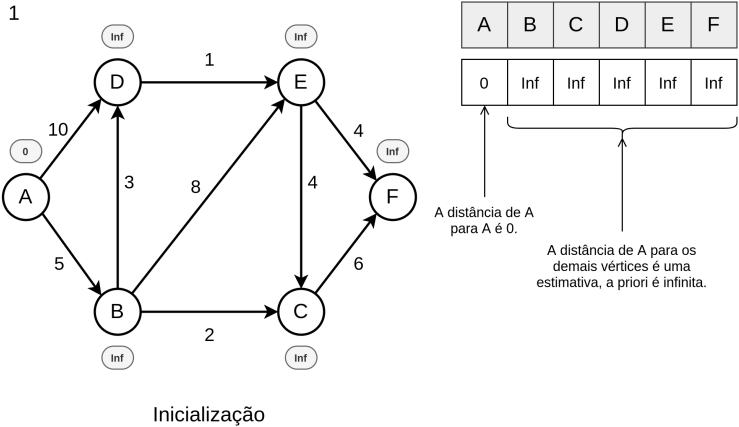


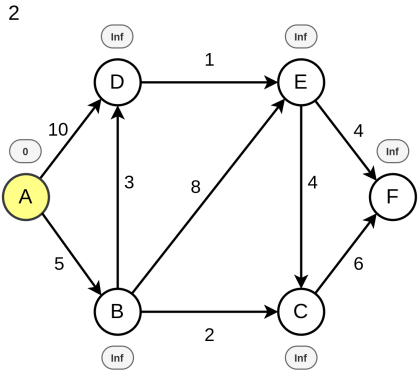
Figura 2: Grafo dirigido ponderado.

0



A	B	C	D	E	F

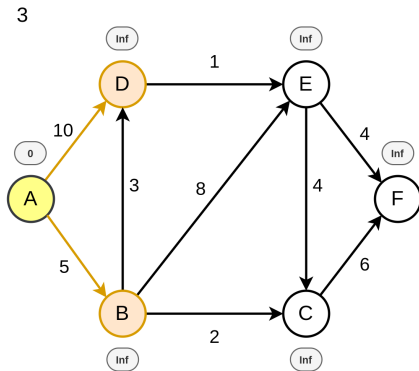




Seleção

A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf

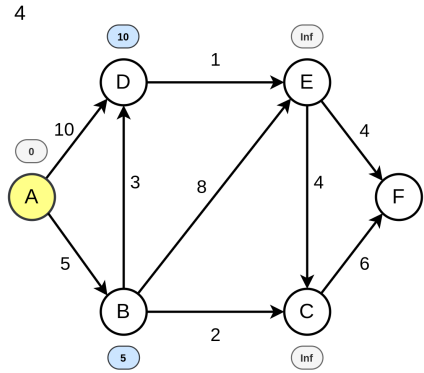
A menor distância dentre os  
vértices não fechados é  
selecionada



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf

As distâncias para os  
vértices adjacentes  
são atualizadas

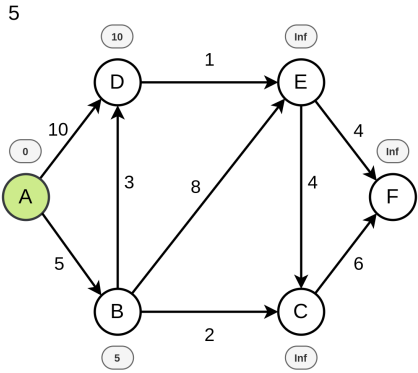




Relaxamento das arestas

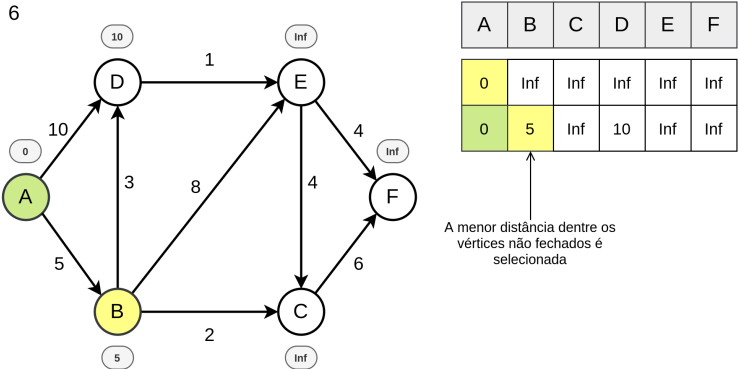
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5<Inf	Inf	10<Inf	Inf	Inf
0	0+5	Inf	0+10	Inf	Inf

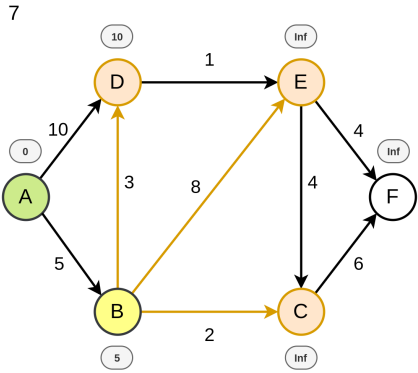
Se as distâncias atuais são menores, atualize elas somando a distância percorrida até o momento



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf

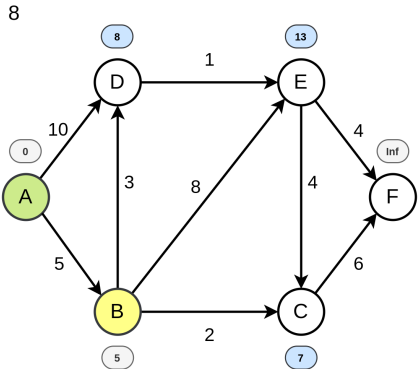
Fechamento do vértice





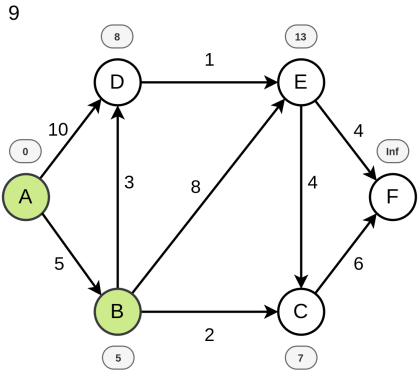
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	2	3	8	Inf

As distâncias para os  
vértices adjacentes  
são atualizadas

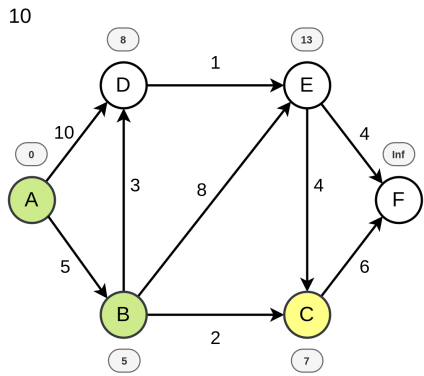


A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	5+2	5+3	5+8	Inf

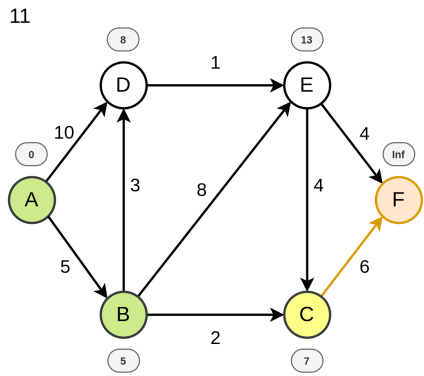
Se as distâncias atuais são menores, atualize elas somando a distância percorrida até o momento



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf

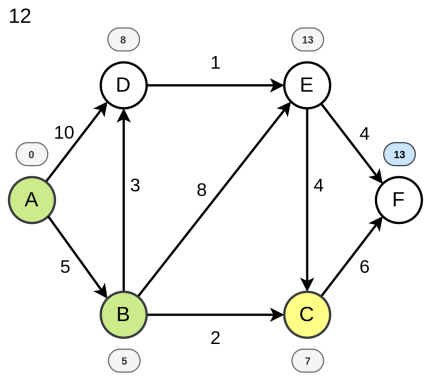


A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf

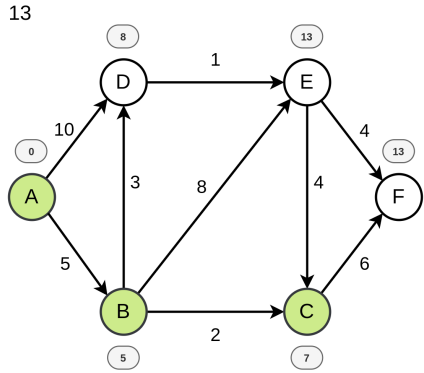


A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	6

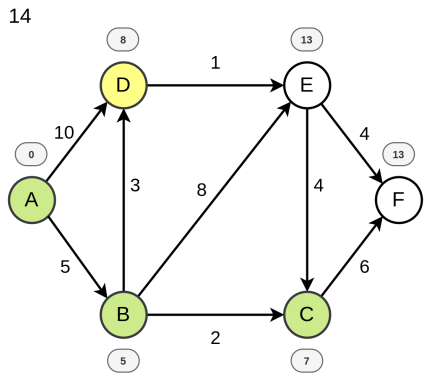




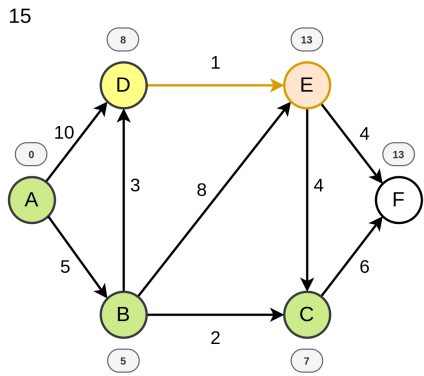
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13<Inf 7+6



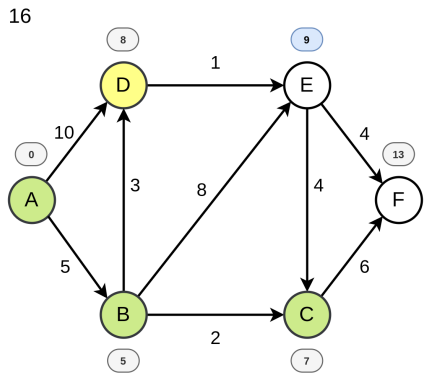
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13



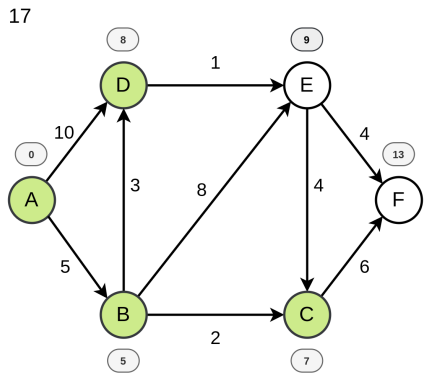
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13



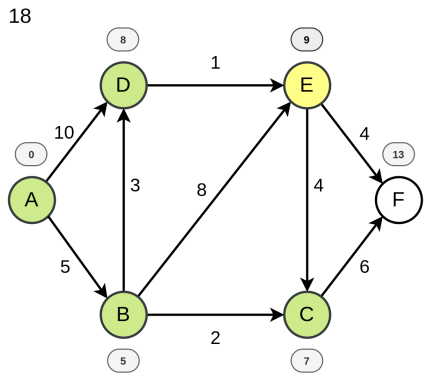
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	1	13



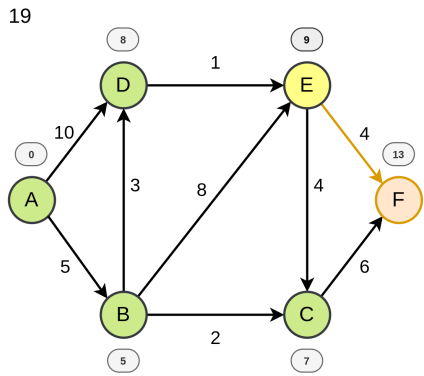
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	8+1	13



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13



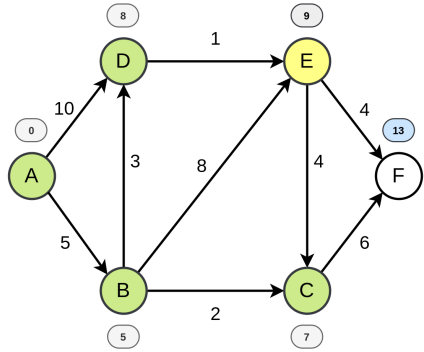
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13
0	5	7	8	9	13

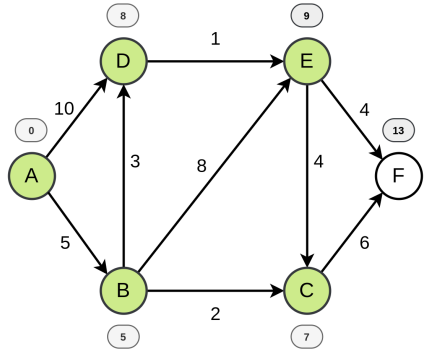


20



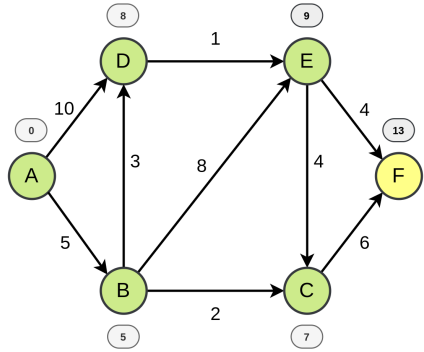
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13
0	5	7	8	9	13

21



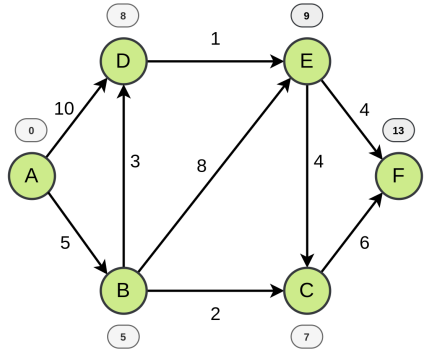
A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13
0	5	7	8	9	13

22



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13
0	5	7	8	9	13

23



A	B	C	D	E	F
0	Inf	Inf	Inf	Inf	Inf
0	5	Inf	10	Inf	Inf
0	5	7	8	13	Inf
0	5	7	8	13	13
0	5	7	8	9	13
0	5	7	8	9	13
0	5	7	8	9	13

```
1 void dijkstra(Graph *g, void *start) {
2     PriorityQueue *openVertices; // Fila de prioridade mínima
3     Vertex *v;
4     Edge *e;
5
6     openVertices = init(g, start);
7
8     while (!PriorityQueue_isEmpty(openVertices)) {
9         v = PriorityQueue_extractMin(openVertices);
10        e = v->first;
11        while (e) {
12            relax(e);
13            e = e->next;
14        }
15    }
16 }
```

- Em um mapa de rotas de trânsito cada cruzamento pode ser mapeado como um vértice e cada rua uma aresta.

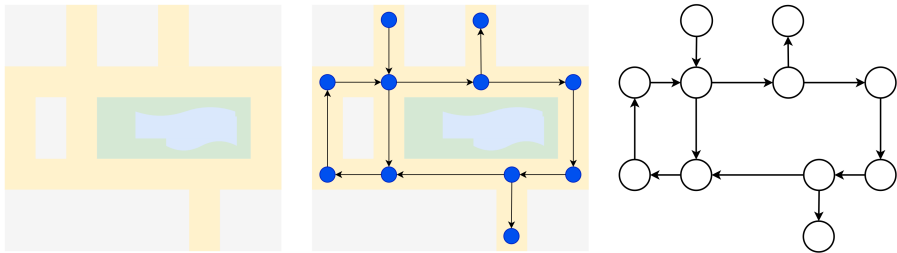


Figura 3: Modelagem de um mapa de trânsito como um grafo.

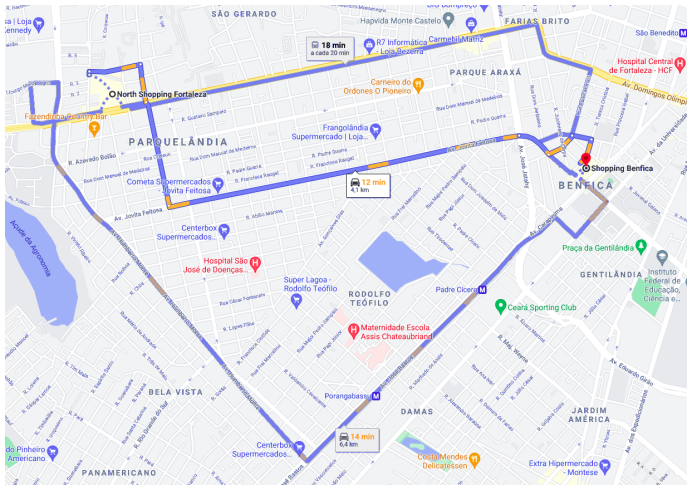


Figura 4: Rotas entre dois pontos com melhor tempo.

### Exercício

Crie uma representação do grafo na figura ao lado. Implemente a função `void dijkstra(...)` que calcula a menor distância entre um vértice e os demais e mostre qual o menor caminho entre os vértices 1 e 12.

