

# Oficina 5 - Calculando as Rotas mais Curtas a partir de Belo Horizonte

Oficina de AEDs2 - prof. Matheus Pereira

## Contextualização

Um sistema de navegação precisa calcular a menor distância para viajar de **Belo Horizonte** até outras cidades conectadas por estradas. Como as estradas podem ter diferentes comprimentos, é importante sempre escolher os caminhos que levam a menores distâncias acumuladas.

Para organizar e atualizar eficientemente as distâncias descobertas durante o processo, o sistema utilizará uma **min-heap**, onde sempre será retirada a cidade mais próxima ainda não processada.

## Descrição do Problema

Você deve implementar um programa que:

1. Leia uma lista de cidades e estradas entre elas, com seus comprimentos (em quilômetros).
2. Considere Belo Horizonte como a cidade inicial.
3. Calcule qual é a menor distância possível para chegar de Belo Horizonte a cada uma das outras cidades.
4. Imprima o nome de cada cidade junto com a sua menor distância encontrada (**opcional: imprima de forma ordenada pela distância utilizando o HeapSort**).

### Observações:

- As estradas ligam duas cidades e podem ser percorridas nos dois sentidos.
- Pode haver mais de um caminho entre Belo Horizonte e outra cidade; deve ser escolhida sempre a menor distância.
- Você deverá **implementar sua própria min-heap** para organizar as cidades de acordo com suas distâncias parciais.

## Formato da Entrada

A primeira linha contém dois números inteiros:

- $N \rightarrow$  número de cidades

- $M \rightarrow$  número de estradas

As próximas  $M$  linhas descrevem as estradas no formato:

CidadeA CidadeB Distancia

## Saída Esperada

O programa deve imprimir a distância mínima entre Belo Horizonte e cada outra cidade:

Cidade: Distancia

## Exemplo de Entrada 1

```
5 6
BeloHorizonte SaoPaulo 586
BeloHorizonte RioDeJaneiro 434
RioDeJaneiro SaoPaulo 429
RioDeJaneiro Vitoria 521
SaoPaulo Curitiba 408
Curitiba Vitoria 1000
```

## Exemplo de Saída 1

```
Curitiba: 994
RioDeJaneiro: 434
SaoPaulo: 586
Vitoria: 955
```

## Exemplo de Entrada 2

```
12 15
BeloHorizonte Contagem 20
BeloHorizonte Betim 31
BeloHorizonte SeteLagoas 74
BeloHorizonte OuroPreto 100
BeloHorizonte Divinopolis 120
Contagem Betim 18
Contagem SeteLagoas 62
Betim Divinopolis 90
SeteLagoas LagoaSanta 47
LagoaSanta Vespasiano 15
Vespasiano Sabara 25
Sabara Caete 20
Caete Mariana 38
OuroPreto Mariana 20
Divinopolis Itauna 45
```

## Exemplo de Saída 2

Betim: 31  
Caete: 165  
Contagem: 20  
Divinópolis: 111  
Itauna: 156  
LagoaSanta: 121  
Mariana: 120  
OuroPreto: 100  
Sabara: 135  
SeteLagoas: 74  
Vespasiano: 136

## Passo a Passo Simplificado

1. Guarde as cidades e estradas.
2. Inicialize todas as distâncias como infinitas, exceto Belo Horizonte que terá valor 0.
3. Use a min-heap para retirar sempre a cidade mais próxima ainda não processada.
4. Atualize as distâncias dos vizinhos e reinsira-os na heap quando necessário.
5. Continue até que a heap esteja vazia.

## Evolução da Min-Heap (Exemplo do primeiro caso)

1. **Inicialização:** Heap = [(0, BeloHorizonte)]  
Distâncias conhecidas: BH = 0, todas as outras cidades = infinito
2. **Passo 1 – Retira Belo Horizonte (0):** Para cada vizinho, calcula-se a distância passando pela cidade atual:
  - SP:  $0 \text{ (distância de BH)} + 586 \text{ (distância BH} \rightarrow \text{SP)} = 586 \rightarrow \text{atualiza}$
  - Rio:  $0 + 434 = 434 \rightarrow \text{atualiza}$

Heap após inserções = [(434, RioDeJaneiro), (586, SaoPaulo)]

3. **Passo 2 – Retira Rio de Janeiro (434):** Atualiza vizinhos considerando a distância acumulada:
  - SP via Rio:  $434 + 429 = 863 \rightarrow \text{não atualiza (já era 586)}$
  - Vitória via Rio:  $434 + 521 = 955 \rightarrow \text{atualiza}$

Heap = [(586, SaoPaulo), (955, Vitoria)]

4. **Passo 3 – Retira São Paulo (586):** Atualiza vizinhos:
  - Curitiba via SP:  $586 + 408 = 994 \rightarrow \text{atualiza}$

Heap = [(955, Vitoria), (994, Curitiba)]

5. **Passo 4 – Retira Vitória (955):** Verifica vizinhos: nenhuma distância é menor que a já conhecida → nenhuma atualização

Heap = [(994, Curitiba)]

6. **Passo 5 – Retira Curitiba (994):** Verifica vizinhos: nenhuma distância é menor → nenhuma atualização

Heap = []

## Evolução da Min-Heap – Exemplo 2 (12 cidades)

1. **Inicialização:**

Heap = [(0, BeloHorizonte)]

Distâncias conhecidas: BH = 0, todas as outras cidades = infinito

2. **Passo 1 – Retira Belo Horizonte (0):** Atualiza vizinhos:

- Contagem:  $0 + 20 = 20 \rightarrow$  atualiza
- Betim:  $0 + 31 = 31 \rightarrow$  atualiza
- SeteLagoas:  $0 + 74 = 74 \rightarrow$  atualiza
- OuroPreto:  $0 + 100 = 100 \rightarrow$  atualiza
- Divinópolis:  $0 + 120 = 120 \rightarrow$  atualiza

Heap = [(20, Contagem), (31, Betim), (74, SeteLagoas), (100, OuroPreto), (120, Divinópolis)]

3. **Passo 2 – Retira Contagem (20):** Atualiza vizinhos:

- Betim via Contagem:  $20 + 18 = 38 \rightarrow$  não atualiza (já era 31)
- SeteLagoas via Contagem:  $20 + 62 = 82 \rightarrow$  não atualiza (já era 74)

Heap = [(31, Betim), (74, SeteLagoas), (100, OuroPreto), (120, Divinópolis)]

4. **Passo 3 – Retira Betim (31):** Atualiza vizinhos:

- Divinópolis via Betim:  $31 + 90 = 121 \rightarrow$  não atualiza (já era 120)

Heap = [(74, SeteLagoas), (100, OuroPreto), (120, Divinópolis)]

5. **Passo 4 – Retira SeteLagoas (74):** Atualiza vizinhos:

- LagoaSanta via SeteLagoas:  $74 + 47 = 121 \rightarrow$  atualiza

Heap = [(100, OuroPreto), (120, Divinópolis), (121, LagoaSanta)]

6. **Passo 5 – Retira OuroPreto (100):** Atualiza vizinhos:

- Mariana via OuroPreto:  $100 + 20 = 120 \rightarrow$  atualiza

Heap = [(120, Divinópolis), (121, LagoaSanta), (120, Mariana)]

7. **Passo 6 – Retira Divinópolis (120):** Atualiza vizinhos:
  - Itauna via Divinópolis:  $120 + 45 = 165 \rightarrow$  atualiza
$$\text{Heap} = [(120, \text{Mariana}), (121, \text{LagoaSanta}), (165, \text{Itauna})]$$
8. **Passo 7 – Retira Mariana (120):** Atualiza vizinhos:
  - Caete via Mariana:  $120 + 38 = 158 \rightarrow$  atualiza
$$\text{Heap} = [(121, \text{LagoaSanta}), (165, \text{Itauna}), (158, \text{Caete})]$$
9. **Passo 8 – Retira LagoaSanta (121):** Atualiza vizinhos:
  - Vespasiano via LagoaSanta:  $121 + 15 = 136 \rightarrow$  atualiza
$$\text{Heap} = [(136, \text{Vespasiano}), (165, \text{Itauna}), (158, \text{Caete})]$$
10. **Passo 9 – Retira Vespasiano (136):** Atualiza vizinhos:
  - Sabara via Vespasiano:  $136 + 25 = 161 \rightarrow$  atualiza
$$\text{Heap} = [(158, \text{Caete}), (165, \text{Itauna}), (161, \text{Sabara})]$$
11. **Passo 10 – Retira Caete (158):** Atualiza vizinhos:
  - Mariana via Caete:  $158 + 38 = 196 \rightarrow$  não atualiza (já era 120)
$$\text{Heap} = [(161, \text{Sabara}), (165, \text{Itauna})]$$
12. **Passo 11 – Retira Sabara (161):** Atualiza vizinhos:
  - Caete via Sabara:  $161 + 20 = 181 \rightarrow$  não atualiza (já era 158)
$$\text{Heap} = [(165, \text{Itauna})]$$
13. **Passo 12 – Retira Itauna (165):** Nenhuma atualização  

$$\text{Heap} = []$$