

Caminhos mínimos em grafos com pesos

Algoritmos e Estruturas de Dados - 2025/02

Última atualização: 25/07/2025 11:03

Simule a ideia seguinte com todos os grafos em cada exercício. Iremos usar essas simulações para construir o algoritmo de Dijkstra.

1. A partir da fonte, selecione iterativamente o vértice não visitado K com menor distância até a fonte
2. para todos os vizinhos de K , atualize suas distâncias até a fonte se o caminho que passa K for melhor que o caminho já encontrado
3. marque K como visitado
4. continue até não haver mais vértices não visitados ou até chegar ao destino

Você deve usar as seguintes estruturas auxiliares:

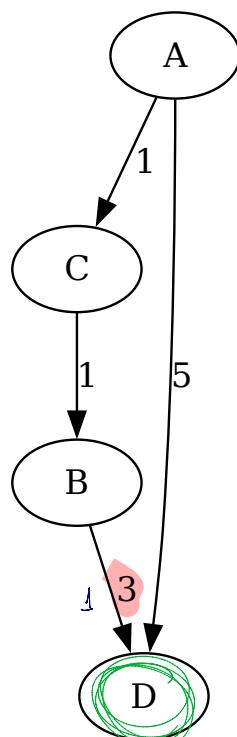
- dist - array com distâncias até a fonte
- pred - array com predecessores no menor caminho até o elemento
- h - conjunto com vértices já visitados

Fonte: A, **Destino:** B

$$\text{DIST} = [0 \ 2 \ 1 \ 5]$$

$$\text{PRED} = [0 \ 2 \ 0 \ 0]$$

h



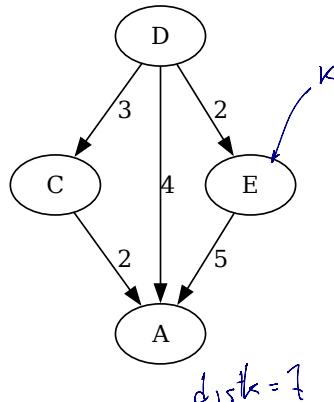
Fonte: D, **Destino:** A

$$\begin{matrix} A & C & D & E \\ 4 & 3 & 0 & 2 \end{matrix}$$

$$\text{DIST} = [4 \ 3 \ 0 \ 2]$$

h

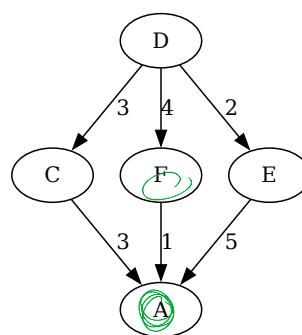
↳ próximo \Rightarrow elemento em h
com menor dist



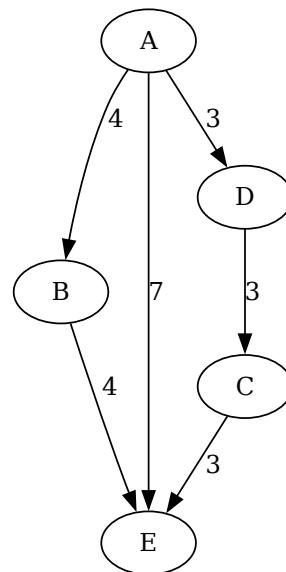
Fonte: D, **Destino:** A

$$\text{DIST} = \begin{matrix} A & C & D & E & F \\ 6 & 3 & 0 & 2 & 4 \end{matrix}$$

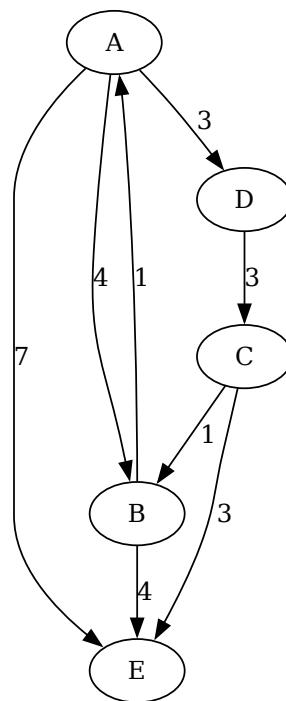
$$\text{PRED} = [1 \ 2 \ 2 \ 2 \ 2]$$



Fonte: A, **Destino:** E



Fonte: A, **Destino:** B

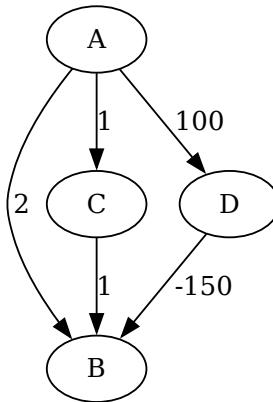


Atenção

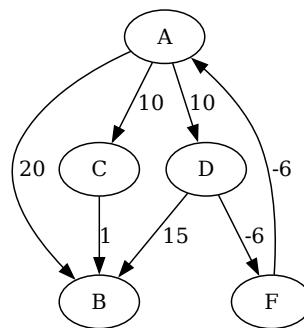
Os grafos abaixo tem pesos negativos! Isso por enquanto não estava nas nossas definições (toda aresta tem peso maior que zero). Vamos só ver o que acontece caso isso exista.

Fonte: A, **Destino:** B

A	B	C	D	
PRED :	0	-1	-1	-1
DIST :	0	+∞	+∞	+∞

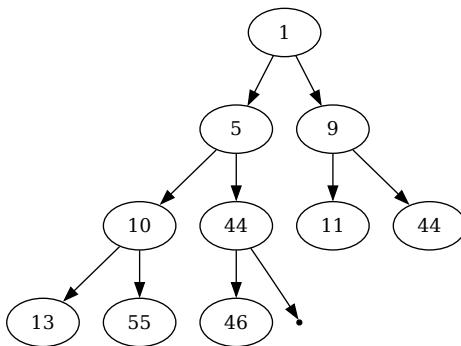


Fonte: A, **Destino:** B

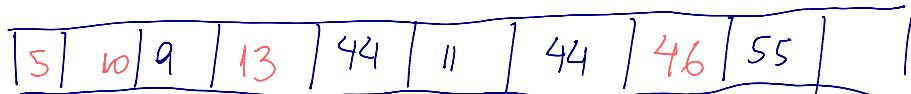


Min-heap

Considere o heap abaixo para os próximos 3 exercícios.



Exercício: Escreva a representação em array do heap acima



Exercício: Execute MENOR no heap, desenhe-o abaixo e depois escreva sua representação em array.

$$N \leftarrow h.length$$

$$RES \leftarrow h[0]$$

$$h[0] \leftarrow h[N-1]$$

$$N \leftarrow N-1$$

$$atual \leftarrow 0$$

while $2 * atual + 1 < N$ do

$$\text{menor} \leftarrow 2 * atual + 1$$

if $2 * atual + 2 < N$ AND $h[2 * atual + 2] < h[\text{menor}]$

$$\text{menor} \leftarrow 2 * atual + 2$$

end

if $h[atual] > h[\text{menor}]$ then

$$\text{troca } h[atual] \leftarrow h[\text{menor}]$$

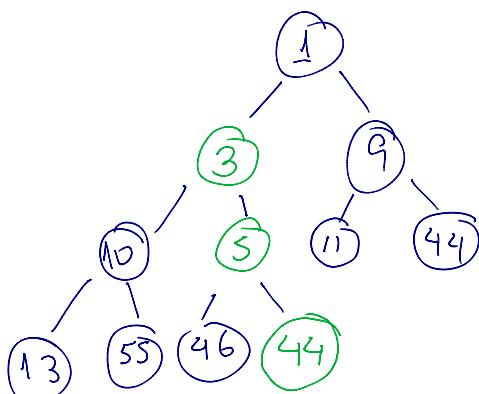
$$atual \leftarrow \text{menor}$$

menor filho

else

. break

Exercício: Execute INSERE(3) no heap acima, desenhe-o ao lado e escreva sua representação em array.



$$N \leftarrow h.length$$

$$h[N] \leftarrow v$$

while $h[N] < h[(N-1)/2]$ do

TROCA $h[N]$ e $h[(N-1)/2]$

$$N \leftarrow (N-1)/2$$

end