

Descrição

O Algoritmo de busca por Árvore Geradora (BFS: Breath-first search)

Input

G grafo conexo com n vértices v_1, \dots, v_n

Output

uma árvore geradora T

Pseudocódigo

```
bfs(V,E){
    // V = vértices ordenadas  $v_1, \dots, v_n$  de  $G$ 
    // E = arestas
    // V' = vértices da árvore geradora T
    // E' = arestas da árvore geradora T
    //  $v_1$  = raiz de T
    // S = lista ordenada

    S = ( $v_1$ )
    V' = { $v_1$ }
    E' = NULL

    While(true){
        for each X  $\in$  S, in order,
            for each y  $\in$  V\V', in order,
                f(x,y)  $\in$  E'  $\cup$  {(x,y)}, V' = V'  $\cup$  {V}
            if no edges are added
                return T
            S = children of S ordered
    }
}
```

Inicialização

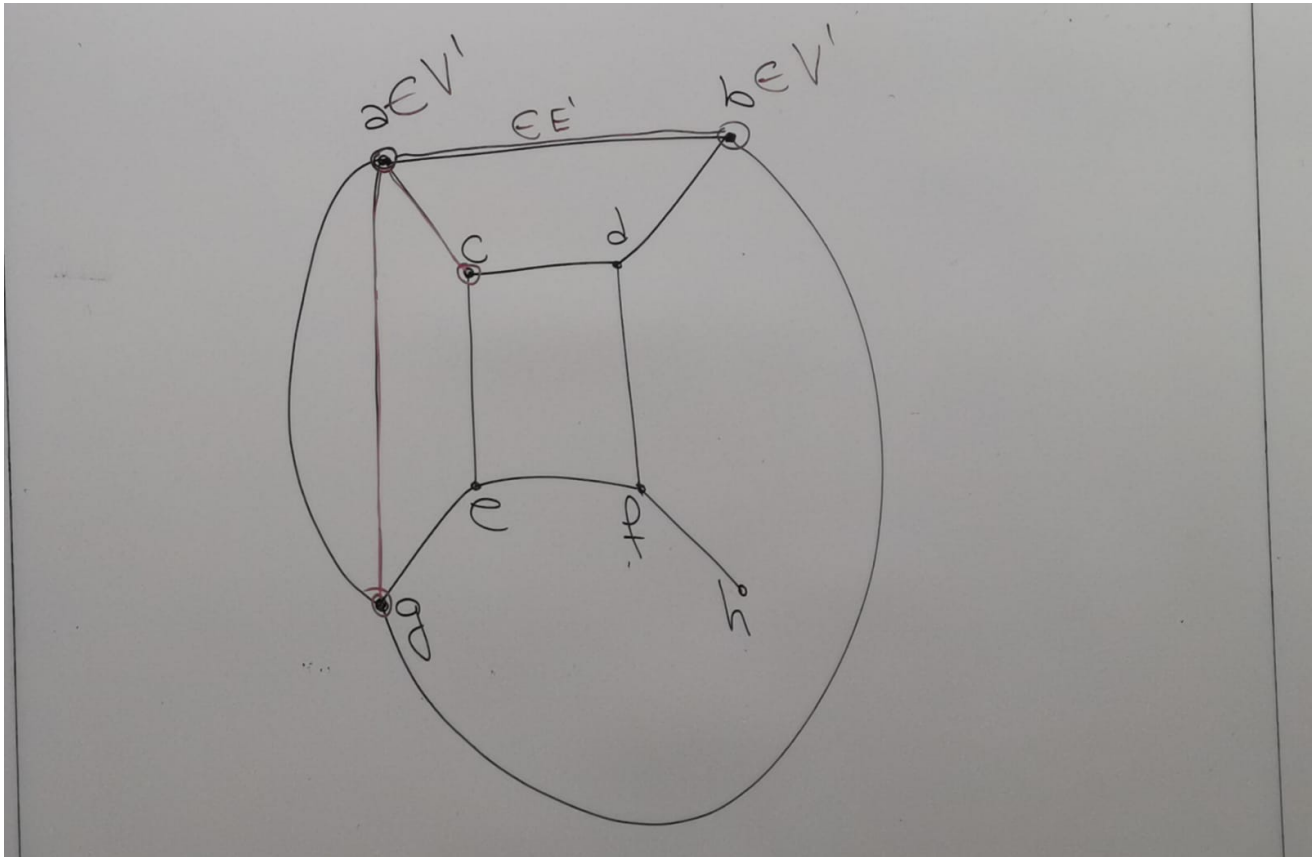
- Declare uma lista S e insira o vértice inicial v_1 .
- Inicialize o array de vértices visitados V' e marque o vértice inicial v_1 como visitado.
- Inicialize o array de arestas A' como nulo.

Iterações

Siga o processo até a lista ficar vazia

- Remove o primeiro vértice da lista.
- Marcar esse vértice como visitado.
- Inserir todos os vizinhos não-visitados desse vértice na lista de vértices visitados V' .

Exemplo:



Lista Ordenada de vértices:

(a, b, c, d, e, f, g, h)

$S = (a)$

$V' = \{a\}$

$E' = \phi$

$V' = \{a, b\}$

$E' = \{(a, b)\}$

$V' = \{a, b, c\}$

$E' = \{(a, b), (a, c)\}$

$V' = \{a, b, c, d\}$

$E' = \{(a, b), (a, c), (a, g)\}$

$S = (b, c, g)$ //filhos de S

a (yes)

b

$V' = \{a, b, c, g, e\}$

$$E' = \{(a, b), (a, c), (a, g), (b, d)\}$$

c

$$V' = \{a, b, c, g, d, e\}$$

$$E' = E' \cup (c, e)$$

g

$$S = (d, e)$$

$$V' = V' \cup f = \{a, b, c, d, g, d, e, f\}$$

$$E' = E' \cup (d, f)$$

$$S = (f)$$

$$V' = V' \cup h = V$$

$$E' = E' \cup (f, h)$$

$$S = (h)$$

Output: árvore geradora $T' = (V', E')$