

Grafos

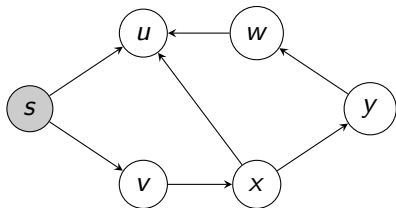
Ricardo Dutra da Silva

Universidade Tecnológica Federal do Paraná

Algoritmo: BuscaProfundidade($G = (V, E), v$)

```
1  $v.visitado \leftarrow \text{verdadeiro}$ 
2 para  $(v, w) \in E$  faça
3   se  $w.visitado = \text{falso}$  então
4     BuscaProfundidade( $G, w$ )
```

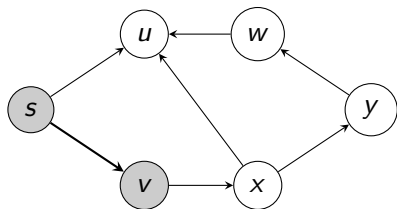
Busca em Profundidade



Pilha de recursão

`BuscaProfundidade(G, s)`

Busca em Profundidade

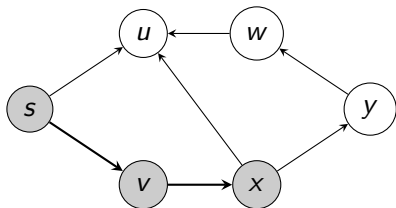


Pilha de recursão

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

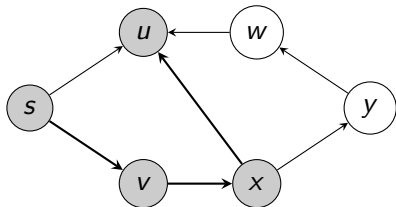
Busca em Profundidade



Pilha de recursão

BuscaProfundidade(G, x)
BuscaProfundidade(G, v)
BuscaProfundidade(G, s)

Busca em Profundidade



Pilha de recursão

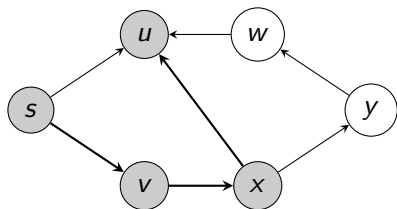
BuscaProfundidade(G, u)

BuscaProfundidade(G, x)

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

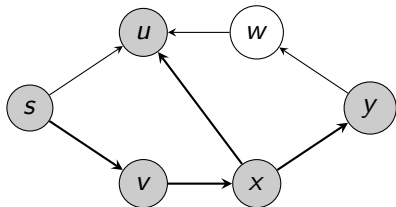
Busca em Profundidade



Pilha de recursão

BuscaProfundidade(G, x)
BuscaProfundidade(G, v)
BuscaProfundidade(G, s)

Busca em Profundidade



Pilha de recursão

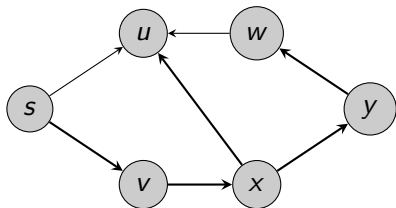
BuscaProfundidade(G, y)

BuscaProfundidade(G, x)

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

Busca em Profundidade



Pilha de recursão

BuscaProfundidade(G, w)

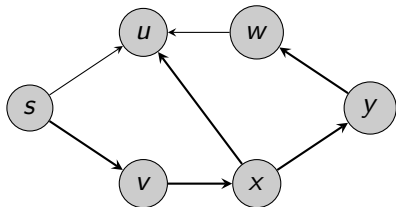
BuscaProfundidade(G, y)

BuscaProfundidade(G, x)

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

Busca em Profundidade



Pilha de recursão

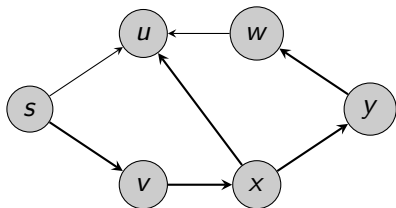
BuscaProfundidade(G, y)

BuscaProfundidade(G, x)

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

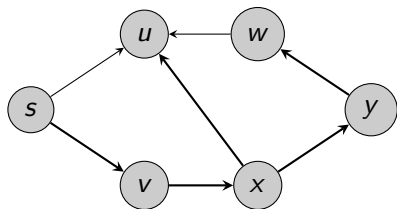
Busca em Profundidade



Pilha de recursão

BuscaProfundidade(G, x)
BuscaProfundidade(G, v)
BuscaProfundidade(G, s)

Busca em Profundidade

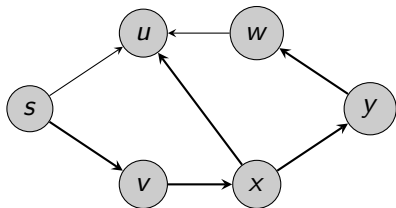


Pilha de recursão

BuscaProfundidade(G, v)

BuscaProfundidade(G, s)

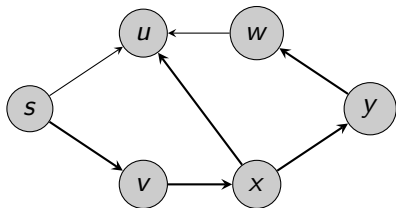
Busca em Profundidade



Pilha de recursão

`BuscaProfundidade(G, s)`

Busca em Profundidade



Pilha de recursão

Análise da complexidade do algoritmo (supondo um grafo conexo):

- linhas 1 executada uma vez para cada vértice, $\mathcal{O}(n)$;
- linhas 2 e 3 executadas uma vez para cada aresta, $\mathcal{O}(m)$ vezes;
- linha 4 tem execução limitada pelas linhas 2 e 3, $\mathcal{O}(m)$ vezes.

Portanto, o algoritmo tem complexidade $\mathcal{O}(n + m)$.

