Grafos

Ricardo Dutra da Silva

Universidade Tecnológica Federal do Paraná

Entrada

Um grafo $G = (V, E)$ e um vértice de início $s \in V$.

Saída

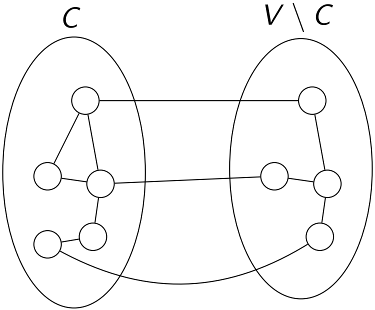
Explorar $v \in V$ se e somente existe um caminho $s \rightsquigarrow v$.

O objetivo é encontrar todos o vértices que possuem um caminho a partir de um vértice de início.

Para eficiência não queremos visitar/explorar um mesmo vértice mais de uma vez.

Definição

Uma aresta (u, v) *de um corte* C *é uma aresta em que* $u \in C$ *e* $v \in V \setminus C$.



Algoritmo: Busca($G = (V, E)$, s)

```
para v ∈ V faça
    v.visitado ← falso
s.visitado ← verdadeiro
/* Escolhe aresta (u, v) ∈ E com u visitado e v não visitado.
   Retorna se achou ou não.                                  */
existe_aresta ← EscolheAresta(u,v)
enquanto existe_aresta faça
    v.visitado ← verdadeiro
    existe_aresta ← EscolheAresta(u,v)
```

Busca Genérica

Algoritmo: RTRANSPOSTA(X, n)

Entrada: Matriz X de dimensão $n \times n$

Saída: Matriz transposta $Y = X^T$

```
se n = 1 então
    retorna Y[1,1] ← X[1,1]
Particiona X em quatro submatrizes X11, X12, X21 e X22, todas
    de tamanho (n/2) × (n/2)
Y11 ← RTRANSPOSTA(X11, n/2)
Y21 ← RTRANSPOSTA(X12, n/2)
Y12 ← RTRANSPOSTA(X21, n/2)
Y22 ← RTRANSPOSTA(X22, n/2)
Concatena as submatrizes Y11, Y12, Y21 e Y22 em uma matriz
    Y de tamanho n × n
retorna Y
```

Demonstração.

Vamos supor inicialmente $v \in V$ visitado e mostrar que existe $s \rightsquigarrow v$ e mostrar por indução que isso é verdade para todo o algoritmo.
Consideraremos que no algoritmo todo vértice visitado pertence a um conjunto C e todo vértice não visitado está no conjunto $V \setminus C$.
Temos portanto um corte no grafo.
Base.
Usamos o fato de que o primeiro vértice em C é s e trivialmente existe $s \rightsquigarrow s$.

□

Demonstração.

Passo Indutivo.
Pela hipótese indutiva temos que, em uma interação qualquer do algoritmo, todo $u \in C$ exite $s \rightsquigarrow u$.
Agora, para uma interação específica, o algoritmo escolhe uma aresta $(u, v) \in E$ com $u \in C$ e $v \in V \setminus C$. Como existe $s \rightsquigarrow u$, então a inclusão da aresta (u, v) em $s \rightsquigarrow u$ forma um caminho $s \rightsquigarrow v$. Logo, v visitado e existe $s \rightsquigarrow v$. $\square$

Busca Genérica

Demonstração.

Vamos supor agora que existe $s \rightsquigarrow v$ e que como consequência v foi visitado.

Por contradição, vamos supor que existe $s \rightsquigarrow v$ mas v não foi visitado ao final do algoritmo.Então podemos considerar um caminho como o abaixo.

Demonstração.

A única maneira de v não ter sido visitado é se existe uma aresta (u, w) em que u foi visitado mas w não foi visitado pelo algoritmo. Possivelmente $u = s$ e $w = v$.
No entanto, o laço do algoritmo não terminaria se existisse tal aresta, contradizendo nossa suposição inicial. Portanto, v foi visitado. $\square$

A análise de tempo deste algoritmo depende do algoritmo para escolha de aresta.

Vamos ver duas opções de escolher arestas que levam a algoritmos eficientes para a busca em grafos: busca em largura e busca em profundidade.