

AULA 08 – PONTOS DE ARTICULAÇÃO, PONTES E COMPONENTES BICONEXOS: EXERCÍCIO

Prof. Daniel Kikuti

Universidade Estadual de Maringá

15 de abril de 2015

PONTOS DE ARTICULAÇÃO, PONTES E COMPONENTES BICONEXOS

Definições

Seja $G = (V, E)$ um grafo **não direcionado conexo**

- ▶ Um **ponto de articulação** ou **vértice de corte** de G é um vértice cuja remoção desconecta G .
- ▶ Uma **ponte** de G é uma aresta cuja remoção desconecta G .
- ▶ Um grafo G é **biconexo** ou **2-conexo** se é conexo e, mesmo com a remoção de um vértice, o grafo permanece conexo.
- ▶ Um **componente biconexo** é um subgrafo biconexo maximal.

Exemplo

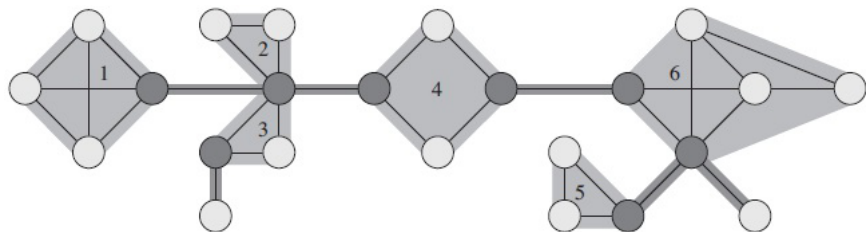


Figura 22.10 [Cormen]

Desafio

Problemas

Para um grafo conexo e não orientado G :

- ▶ Encontre todos os pontos de articulação de G .
- ▶ Encontre todas as pontes de G .
- ▶ Informe se G é biconexo.

Inspiração

Algoritmo de busca em profundidade.

Pontes em grafos

Relembrando definição

Uma **ponte** é uma aresta cuja remoção aumenta o número de componentes do grafo.

Problema

Encontrar as pontes de um grafo dado.

Solução trivial

Aplicar cegamente a definição de ponte. Qual a complexidade disto?

Pontes em grafos

Fato importante

Uma aresta é uma ponte se e somente se ela não pertence a um ciclo não trivial (ciclo com comprimento 2).

Propriedade

Em qualquer árvore de busca em profundidade, uma aresta (v, w) é uma ponte se e somente se não há nenhuma aresta de retorno que conecta um descendente de w a um ancestral de w .

Demonstração

Se há tal aresta, (v, w) não pode ser uma ponte. Por outro lado, se (v, w) não é uma ponte, então tem de haver algum caminho de w a v no grafo diferente do próprio caminho (w, v) . Este caminho deve conter tal aresta.

Pontes em grafos

Definindo `ord` e `low`

- ▶ Para cada vértice v , `ord`[v] representa o número de pré-ordem (apenas tempo de descoberta).
- ▶ `low`[v] é o menor número de pré-ordem que pode ser alcançado a partir de v usando qualquer aresta de retorno, exceto a aresta $(v, \text{pred}[v])$.

Reformulando a propriedade

Em qualquer floresta DFS de um grafo, uma aresta (v, w) da floresta é uma ponte se e somente se `low`[w] é igual a `ord`[w]. Assim, podemos encontrar todas as pontes de um grafo com uma única busca em profundidade.

Pontes em grafos

```
static int conta, ord[maxV], numpts, low[maxV];
static Vertex pred[maxV];

int GRAPHbridges( Graph G) {
    Vertex v;
    conta = numpts = 0;
    for (v = 0; v < G->V; v++)
        ord[v] = -1;
    for (v = 0; v < G->V; v++)
        if (ord[v] == -1) {
            pred[v] = v;
            bridgeR( G, v);
        }
    return numpts;
}
```


Pontes em grafos

```
void bridgeR( Graph G, Vertex v) {
    link a;
    ord[v] = conta++;
    low[v] = ord[v];
    for (a = G->adj[v]; a != NULL; a = a->next) {
        Vertex w = a->w;
        if (ord[w] == -1) {
            pred[w] = v;
            bridgeR( G, w);
            if (low[w] < low[v]) low[v] = low[w];
            if (low[w] == ord[w]) {
                numpts++;
                printf( "%d-%d\n", v, w);
            }
        }
        else if (w != pred[v] && low[w] < ord[v])
            low[v] = ord[w];
    }
}
```

Articulações

Relembrando definição

Uma **articulação** é um vértice cuja remoção aumenta o número de componentes do grafo.

Problema

Encontrar as articulações de um grafo dado.

Solução trivial

Aplicar cegamente a definição de articulação. Qual a complexidade disto?

Articulações

Exercício importante

Desenvolva um algoritmo que encontra pontos de articulação de um grafo G . Analise sua complexidade.

Exercício

[Cormen 22-2] Verificar no livro (exercício difícil).