Teoria dos Grafos
Prova 1 - Shape Aluno

André Thiago Borghi Couto[1]
São Mateus - ES, Brasil

---

1. Seja G um grafo, T uma Arvore geradora de G e $v \in V(G)$.

1.1. E verdade que se v tem grau maior que 1 em T, entao v é vértice de corte em G? Justifique.

R: Não, pode haver outro caminho que liga o acestral de v a seus decendentes, como segue no contra exemplo. Apesar do vértice ter grau $2 > 1$ em T este não é vertice de corte em G, uma vez que o grafo $G - v$ não possui mais componentes conexas que G

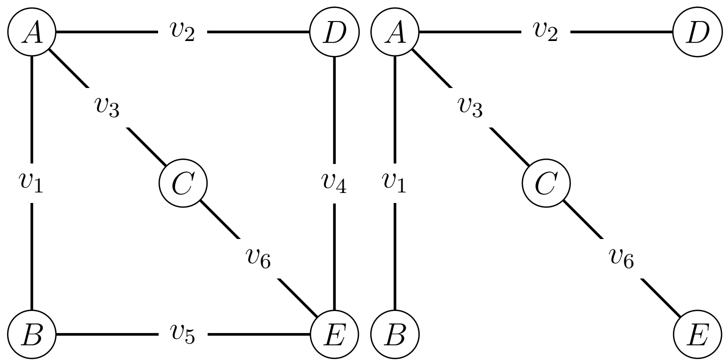

Figura 1: Grafos usados na resposta da questão 1

1.2. E verdade que se v é vértice de corte em G então v tem grau maior que 1 em T? Justifique.

R: Sim, como v é um vértice de corte em G, isto significa que existe um ou mais vértices que apenas são alcançáveis em G a partir de v. Logo uma árvore geradora T não poderá ter

Email address: `andreww.max@hotmail.com` (André Thiago Borghi Couto)
URL: `github/andrewwmax` (André Thiago Borghi Couto)
[1]Estudante de Engenharia da Computação, na Universidade Federal do Espírito Santo no campus Ceunes

v, com grau 1 (folha), pois caso isto ocorra ou se existir algum vértice u em G que não está em T, e portanto T não seria uma árvore geradora de G, ou existir algum outro caminho em G que liga um ancestral de v a algum descendente u. Em ambos os casos teríamos um absurdo, uma vez que T é uma árvore geradora de G e v é vértice de corte em G.

1.3. Um vértice de corte em um grafo pode ser folha de uma árvore geradora deste grafo? Justifique.

R: Não, como visto no item anterior (b), isto não seria possível, uma vez que em árvores T todas as folhas tem grau 1 e como visto que para uma árvore geradora T de G, vértices de corte de G não podem ser folhas de T, segue um contra exemplo para complementar. Em nenhuma delas v possui $grau = 1$ ou seja, v não é folha em nenhuma das árvores geradoras.

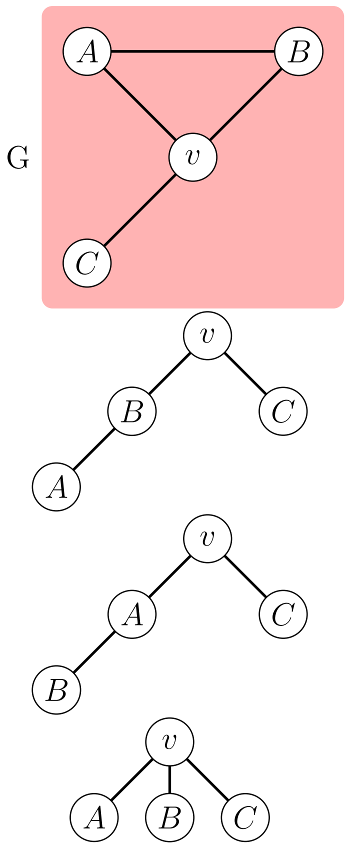

Figura 2: Grafos usados na resposta da questão 1

2. Seja $\Delta(T)$ o maior grau de vértice de uma árvore T. Prove que toda arvore T tem (pelo menos) $\Delta(T)$ folhas.

Dado uma árvore T de n vértices e seja v o vértice de maior grau de T, isto indica que existem $\Delta(T)$ arestas que incidem sobre v e também sobre outros u_i vértices, para i variando de T a $\Delta(T)$, cada um destes vértices u_i deverão fazer parte de pelo menos um caminho que leve de v até uma folha distinta, sendo no mínimo $\Delta(T)$ folhas; Suponha que a árvore T não possua menos de $\Delta(T)$ folhas, isto indica que um dado u_i e u_j levam a mesma folha f, o que só seria possível caso exista mais de um caminho entre v e f, um passando por u_i e outro por u_j (que são adjacentes a v) e portanto um ciclo. O que é um absurdo uma vez que T é uma árvore. Logo o número de folhas em T deve ser de no mínimo $\Delta(T)$

3. Sobre o algoritmo de Prim, responda:

3.1. Qual é o objetivo do algoritmo?

R: O algoritmo de Prim recebe um grafo conexo de G e retorna uma árvore geradora de G que possua o valor mínimo para o somatório de custo de suas arestas.

3.2. Enumere os passos do algoritmo.

R: Primeiro: Recebe os parâmetros, grafo $G(V, E)$, custo das arestas w e raiz r;

Segundo: Cria e inicializa o vetor de distâncias d, que tem tamanho v e valores iniciais infinito e o vetor π de predecessores, também de tamanho v e valores iniciais nulos;

Terceiro: Modifica o valor de r no vetor d para 0;

Quarto: Inicializa um veotr Q com os vértices de G, V;

Quinto: Verifica se o vetor Q é vazio, se sim vai para o passo 6, caso contrario vai para o passo 9;

Sexto: Extrai o vértice v de Q, que possui menos valor em d

Sétimo: Para cada u adjacente de v em G, verifica se o custo da aresta (v, u) que é dada por w é menor que o valor que está em $d[u]$. Exceto o passo 8 para todos os (v, u) satisfazem a condição e depois retorna ao passo 5;

Oitavo: Troca o valor em $d[u]$ pelo valor dado por $W(v, u)$;

Nono: Fim do algoritmo, a árvore de custo total mínimo pode ser montada utilizando os valores no vetor de predecessores π e os custos são dados pelo vetor d.

4. Prove que se T é a arborescência resultante de uma busca em largura a partir do vértice r em um grafo conexo G, então o caminho de r a v em T é um caminho mínimo em G para todo $v \in V(G)$.

Dado um grafo conexo $G(V, E)$ e T árvore resultante de uma busca em largura em G o caminho da raiz r de T a qualquer vértice v de T será mínimo, ou seja, conterá o menor número de vértices possíveis, devido ao processo de percurso em largura, onde partindo do vértice r e utilizando uma fila para auxiliar, este colocará todos os vértices u_i, para i variando de 1 ao número de adjacentes de r na fila atraz de r, adicionando as i arestas (r, u_i) a árvore resultante T que era inicialmente vazia e assim removendo da fila, repetindo este processo para os primeiros vértices da fila e nunca adiccionado um vértice u a fila que já exista em T, e nenhuma aresta em T que ligue um par de vértices já presente em T. Obteremos, quando a fila estiver vazia, a árvore que possui o menor caminho entre r e um v qualquer, uma vez que para todos os vértices de T, partindo de r, apenas quando estes vértices eram totalmente explorados é que eram retirados da fila. Suponha que para um T resultante deste processo o caminho de rTv para um v qualquer não seja o número, então deve existor um caminho $rTu + uTv$ que seja menor que rTv, porém nesnte caso u teria sido explorado antes de v, aparecendo em sua frente nafila e teria seus caminhos adicionados primeiramente a T, portanto não ocorria. Ou existe um caminho de u a v em G que não foi considerado pelo algoritmo, o que é um absurdo uma vez que este explora totalmente cada vértice que esta na fila. Logo rTv é o caminho mínimo de r a um vértice qualquer de G.