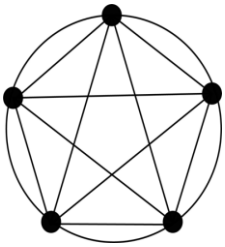
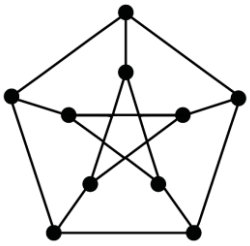
Universidade Federal do Ceará – Campus Russas



Algoritmos em Grafos

Trabalho de Implementação

Prof. Pablo Soares



Bramble Blast(Stickerbush Symphiny)

O trabalho consiste em implementar os algoritmos de **Busca em Profundidade(DFS) & Busca em Largura(BFS)**. Seu algoritmo não deve em hipótese alguma fazer uso de recursão direta ou indireta, ou seja, você deve simular a recursão da busca em profundidade usando uma estrutura de dados adequada.

O seu algoritmo DFS receberá como entrada um grafo $G = (V, E)$ direcionado em forma de arquivo com extensão *.txt* e em seguida deve produzir um arquivo de saída *.txt* com as seguintes informações:

a. Classificação de cada aresta de acordo com o DFS;
b. Ordenação topológica;
c. Quantidade de componentes conexas e quais são elas

O vértice escolhido para iniciar a busca em profundidade será o vértice com maior quantidade de arestas incidentes (saindo e chegando). Em qualquer caso de empate/escolha use o critério de menor *label*. Para o grafo da Figura 01, a busca começaria no vértice **b**.

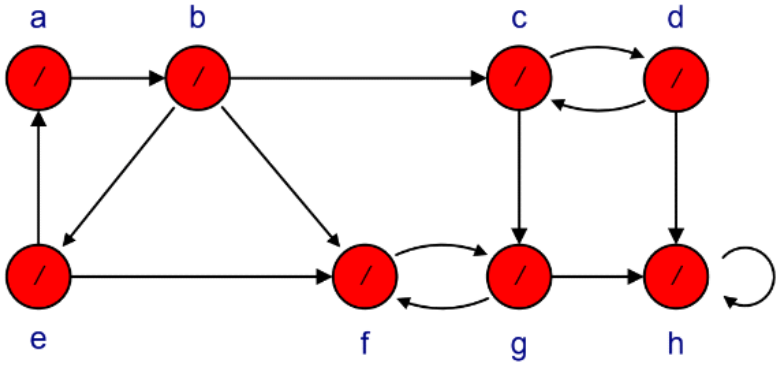


Figura 01: Grafo com $n = 8$

O seu algoritmo BFS receberá como entrada o vértice central k da ordenação topológica obtida na busca em profundidade anterior, onde $k = \lceil n/2 \rceil$ e $n = |V|$. Seu algoritmo deve produzir um outro arquivo de saída *.txt* com as seguintes informações:

a. A menor distância de todos os vértices do grafo com relação ao vértice central *k*;
b. Os vértices que compõem a menor distância de *k* até todos os outros vértices.

Exemplo do arquivo de Entrada.txt do grafo da Figura 01.

A primeira linha do arquivo de entrada contém dois valores inteiros, a quantidade de vértices e a quantidades de arestas do grafo, respectivamente. Da segunda linha em diante, cada linha possui dois valores/*labels* associados a uma aresta do grafo, o vértice de partida e o vértice de chegada.

8 14
a b
b c
b e
b f
c d
c g
d c
d h
e a
e f
f g
g f
g h
h h

Resultado do DFS a partir do vértice **b.**

a b: Retorno
b c: Árvore
b e: Árvore
b f: Avanço
c d: Árvore
c g: Árvore
d c: Retorno
d h: Árvore
e a: Árvore
e f: Cruzamento
f g: Retorno
g f: Árvore
g h: Cruzamento
h h: Retorno
Ordenação Topológica: b e a c g f d h
Componentes Conexas: 4 componentes. [b a e]; [c d]; [g f] [h]

Resultado do BFS a partir do vértice central $\mathbf{c} = \lceil 8/2 \rceil = 4$**.**
a: ∞ → null
b: ∞ → null
c: 0 → c
d: 1 → c – d
e: ∞ → null
f: 2 → c – g – f
g: 1 → c – g
h: 2 → c – d – h

1. Grupo com no máximo 4 alunos;
2. Linguagem de programação “livre”
   - Se for em C → 3.0 pontos na terceira prova
   - Se for em qualquer outra linguagem → 2.0 pontos na terceira prova
   - Se for em Java → 1.0 ponto na terceira prova
3. Defesa do trabalho: Até 14/06/2019