

TRABALHO DE ESTRUTURAS DE DADOS II PARA A SEGUNDA AVALIAÇÃO - PARTE 2

(Equivale a 60% da nota do Trabalho)

- 3) A família de Arthur estão visitando a França, o pai de Arthur, decidiu alugar um carro para conhecer melhor o país. As rodovias são bem cuidadas, todas de sentidos duplo, e duas cidades podem ser ligadas diretamente por mais de uma estrada. No entanto, em todas as estradas paga-se um pedágio de valor fixo (há um pedágio em cada direção, entre duas cidades). O pai de Arthur não tem muito dinheiro para gastar, as viagens com o carro devem ser bem planejadas. Faça um programa em C que dada as cidades e estradas existentes no país, e a cidade onde Arthur e sua família estão, encontre cada cidade (que não a cidade onde eles estão) que possa ser visitada por eles usando busca em profundidade, dada a restrição de que o pai de Arthur deseja pagar no máximo P valor em pedágios (considerando apenas a viagem de ida).

- 4) Refaça o exercício anterior usando agora busca em Largura.

- 5) Compare os resultados das duas soluções, para isso execute o programa com 10 cenários diferentes, e então responda:
- (a) É possível determinar em qual das soluções eles visitara um maior número de cidades? Justifique a resposta.
 - (b) É possível determinar qual solução eles conseguiram economizar mais? Justifique a resposta.
 - (c) Compare o tempo de busca das duas soluções

Obs.: Nos anexos deste documento encontra-se algoritmos de busca de profundidade e largura.

Obs.: No sigaa encontra-se código da busca em largura e profundidade em Python (transforme para C).

Equipe: os programas podem ser feitos em dupla, mas os relatórios são individuais. Se os programas forem feitos em dupla, a dupla deve ser identificada no envio do código.

Data de Entrega: data primeira prova escrita

Entregar: Código Fonte, Relatório(Conforme Modelo em PDF)

Forma de Entrega: pelo SIGAA, caso tenha algum problema enviar por e-mail(julianaoc@fupi.edu.br).

Entrevista Individual: agendar horário com a Professora.

ANEXO A

procedimento Busca(G : Grafo)

Para Cada vértice v de G :

Marque v como não visitado

Para Cada vértice v de G :

Se v não foi visitado:

Busca-Prof(v)

procedimento Busca-Prof(v : vértice)

Marque v como visitado

Para Cada vértice w adjacente a v :

Se w não foi visitado:

Busca-Prof(w)

ANEXO B

procedimento Busca-Largura(v : vértice)

Inicializar F

 Marcar v como visitado

 Colocar v no final de F

Enquanto F não vazio:

$u :=$ primeiro elemento de F

 Retirar u de F

Para cada vértice w adjacente a u :

Se w não foi visitado:

 Marcar w como visitado

 Colocar w no final de F