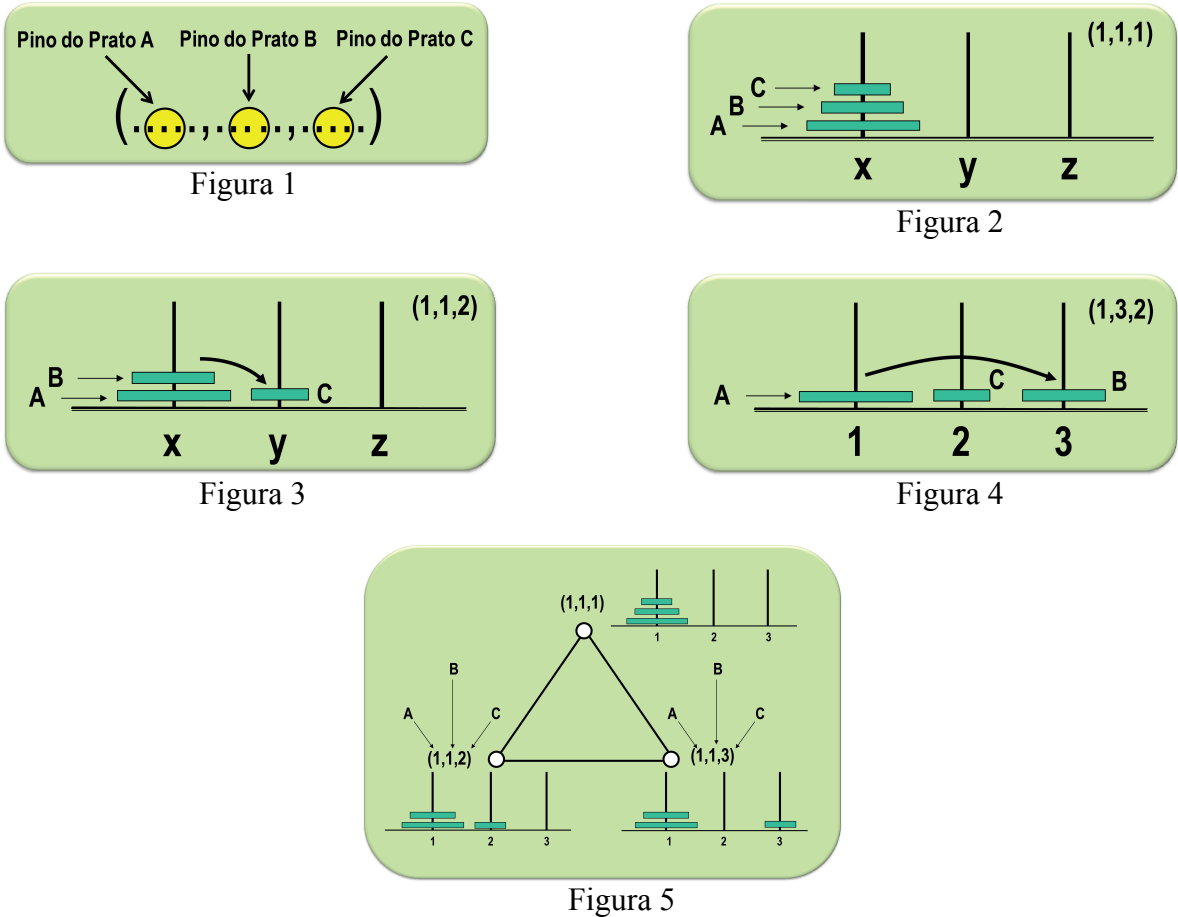
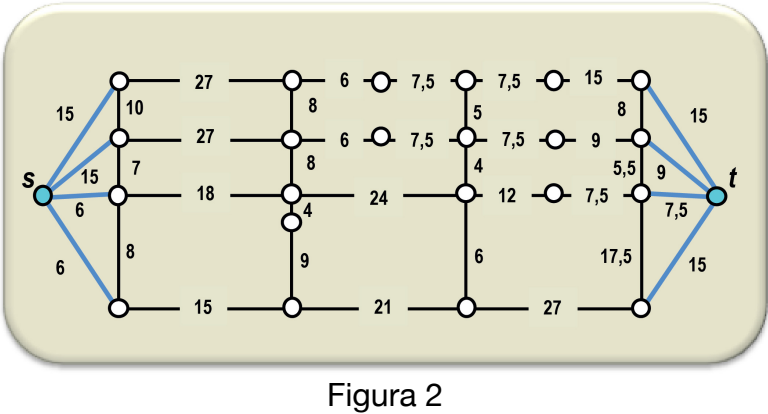


TRABALHO DE ESTRUTURAS DE DADOS II PARA A TERCEIRA AVALIAÇÃO

1) Um problema muito conhecido na ciências da computação é o da Torre de Hanói. O problema clássico da Torre de Hanói em, dados n discos e 3 anos, mover com o menor número possível de movimentos todos os n discos de um suporte origem, de acordo com as seguinte condições: apenas um disco pode ser movimentado de cada vez; os discos movidos deverão ser apoiados sempre em discos de maior tamanho ou na base de algum pino. A Figura 1 exibe os movimentos do desafio da Torre de Hanói. Por convenção uma configuração de discos será representada por um vetor com tantas posições quantas forem os discos. Na posição do disco será marcado o pino onde o disco está assentado, como mostrado na Figuras 2, 3 e 4. A Figura 5 exemplifica como formar um grafo que representa as possibilidade de movimentos dos discos. Cada vértice representa uma diferente configuração do desafio. As configurações são ligadas por arestas se uma configuração pode ser alcançada a partir de outra pelo movimento legal de um disco. De acordo com o exposto, modele o grafo que represente o grafo de movimentos do desafio da Torre de Hanói para o caso de 4 discos. Em seguida crie o grafo usando matriz de adjacência. Depois dada uma determinada configuração dos discos encontre o menor caminho para o resultado final usando o Algoritmo de Ford-Moore-Bellman, para isso coloque valor 1 em cada uma das arestas. Contabilize o tempo gasto para encontrar a solução. (1,5 pontos)



2) Suponha um grafo orientado $G = (V, E)$ no qual cada aresta $(u, v) \in E$ tem um valor associado $r(u, v)$, o qual é um número real no intervalo $0 \leq r(u, v) \leq 1$ que representa a confiabilidade de um canal de comunicação do vértice u até o vértice v. Interpretamos $r(u, v)$ como a probabilidade de que o canal de u até v não venha a falhar, e supomos que essas probabilidades são independentes. Faça um programa eficiente para encontrar o caminho mais confiável entre dois vértices dados. (2,0 pontos)



3) Suponha uma base de dados de 1000 funcionários e que ao invés de ordenar os dados o sistema cria uma tabela hashing para localizar os dados de um funcionário pelo número de matrícula do funcionário. Os dados dos funcionários são: Matrícula, Nome, Função, Salário. A matrícula é uma string de 6 dígitos. Faça um programa para cada um dos itens a seguir que através de uma função hashing e de uma função para colisão organize a base de dados. Depois compare os resultados através do desempenho de cada uma das soluções. E também identifique qual das soluções produz um maior número de colisões. Faça o que se pede considerando um vetor destino de 101 posições e depois com 150 posições. (4,0 pontos)

- (a) Função Hashing: rotação de 2 dígitos para a esquerda depois extrai o 2º, 4º e 6º dígitos e obtenha o resto da divisão pelo tamanho do vetor destino. As colisões devem ser tratadas somando ao resto da divisão o primeiro dígito da matrícula.
- (b) Função Hashing: fole shift com 3 dígitos da seguinte forma: o 1º, 3 e 6º; 2º, 4º e 5º, depois obtenha o resto da divisão do resultado pelo tamanho do vetor destino. As colisões devem ser realizadas somando 7 ao valor obtido.

obs.: Em todos os casos se uma matrícula não conseguir ser colocada porque todos os possíveis locais já estão ocupados, retire a informação da primeira posição encontrada e coloque a nova informação.

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 terceira prova escrita

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

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

Entrevista Individual: agendar horário com a Professora.