

TRABALHO DE ESTRUTURAS DE DADOS II PARA A SEGUNDA AVALIAÇÃO

- 1) **(3,0 pontos)** Faça um programa em C que cadastre em uma árvore vermelha-preta com os dados de:
- (a) Curso: Código, nome do curso, quantidade de blocos, número de semanas para cada disciplina (exemplo: 15, 18, 20. Na UFPI é 15) e o endereço para a árvore vermelha-preta de disciplinas daquele curso. A árvore deve estar organizada pelo código do curso, o mesmo não pode se repetir e cada curso terá uma árvore de disciplinas;
  - (b) Disciplinas: Código da disciplina, nome da disciplina, bloco da disciplina(o qual deve ser menor que a quantidade de blocos do curso), a carga horária da disciplina(deve ser um número múltiplo do número de semanas para cada disciplina no curso). A árvore deve estar organizada pelo código da disciplina, o mesmo não deve se repetir;
  - (c) Depois permita ao usuário:
    - (1) Imprimir a árvore de cursos em ordem crescente pelo código do curso;
    - (2) Imprimir os dados de um curso dado o código do mesmo;
    - (3) Imprimir todos os cursos com a mesma quantidade de blocos, onde o usuário informe a quantidade de blocos;
    - (4) Imprimir a árvore de disciplinas em ordem crescente pelo código das disciplinas dado o código do curso;
    - (5) Imprimir os dados de uma disciplina dado o código dela e do curso ao qual ela pertence;
    - (6) Imprimir as disciplinas de um determinado bloco de um curso, dado o bloco e o código do curso;
    - (7) Imprimir todas as disciplinas de um determinado curso com a mesma carga horária, onde o código do curso e a carga horária devem ser informadas pelo usuário;
    - (8) Excluir uma disciplina dado o código da disciplina e o código do curso;
    - (9) Excluir um curso dado o código do mesmo, desde que não tenha nenhuma disciplina cadastrada para aquele curso.

Faça um experimento que busque por 30 itens(pode ser cursos ou disciplinas), mostre o caminho percorrido na árvore para encontrar o item e o tempo gasto. Depois faça uma análise dos resultados obtidos.

- 2) **(4,0 pontos)** Repita todo o processo do exercício 1, mas agora implemente usando uma Árvore 2-3 em todos os locais que se usa árvore.
- 3) **(3,0 pontos)** Suponha que uma memória seja dividida em blocos lógicos de 1Mbyte, e que o primeiro bloco é o bloco 0, suponha também que o gerenciador de memória de um Sistema Operacional mantenha uma árvore 4-5 com nós para blocos livres e ocupados da memória. Cada nó da árvore 4-5 contém os seguintes campos: O- para ocupado ou L- para livre, o número do bloco inicial, o número do bloco final, endereço inicial (correspondente ao endereço inicial do primeiro bloco do nó) e endereço final (correspondente ao endereço final do último bloco do nó).
- (a) Faça um programa em C que cadastra os nós da árvore, onde o usuário deve informar se o primeiro nó é livre ou ocupado, o endereço inicial e final do nó. Os demais nós serão contabilizados pelo sistema se são livres ou ocupados e o usuário deve apenas informar o endereço final de cada um. O cadastro termina quando o usuário informar como endereço final de um nó o último endereço da memória.
  - (b) Faça uma função que o usuário informe a quantidade de nós que ele precisa alocar e retorne as informações do nó que atenda as necessidades do usuário e então modifique a situação do referido nó de Livre para Ocupado.
    - (i) Lembre-se que a árvore deve manter nós intercalados de acordo com a situação do nó, ou seja, se a situação de um nó muda então os nós adjacentes a ele deve ser concatenados. Consequentemente os nós da árvore serão modificados.

(ii) Lembre-se também se o nó escolhido possui uma quantidade maior de blocos do que o solicitado pelo usuário os nós árvore devem ser atualizados de forma que mantenha blocos adjacentes livres ou ocupados em um mesmo nó.

(c) Faça uma função em que o usuário informe blocos que devem ser liberados.

(i) Lembre-se que a árvore deve manter nós intercalados de acordo com a situação do nó, ou seja, se a situação de um nó muda então os nós adjacentes a ele deve ser concatenados. Consequentemente os nós da árvore serão modificados.

(ii) Lembre-se também se o nó escolhido possui uma quantidade maior de blocos do que o solicitado pelo usuário os nós árvore devem ser atualizados de forma que mantenha blocos adjacentes livres ou ocupados em um mesmo nó.

**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@gmail.com).

**Entrevista Individual:** agendar horário com a Professora.