

Universidade Federal de Santa Catarina – Curso de Sistemas de Informação
Disciplina Estruturas de Dados – Prova 1 – 2021/2 – Turma 03238A

Essa avaliação é composta por 4 questões, cada uma com sub-itens e a pontuação indicada nos mesmos. Leia as questões completamente e com cuidado, assegurando-se de haver entendido o enunciado e o que se pede em cada uma.

Evidentemente, você terá acesso a outros recursos (livros, internet) que lhe permitirão responder por si só cada uma das questões. Responda com honestidade, sem copiar-e-colar, como você entendeu, o que aprendeu. A possibilidade de consulta a outros materiais (e mesmo pessoas) e o prazo dilatado para responder permitirá que você estude, se aprofunde e aprenda o conteúdo e faça relações que talvez não tenha compreendido/heito antes! Excelente! Ajudar a fixar novos conteúdos e identificar os conteúdos mais importantes do conjunto da matéria são objetivos de uma prova, também. Mas escreva com suas palavras, responda por você mesmo, ok?

Prazo: até 23h59 de domingo dia 19 de dezembro.

Forma: responda às questões em um editor de texto e envie o arquivo resultante (de preferência um PDF, mas na impossibilidade, pode ser um .odt ou até mesmo um .docx) ou então responda à mão, em folhas em branco e envie as fotos das respostas (coloque tudo em um arquivo compactado, tipo .ZIP, .tar.gz, ou outro). Se necessário, use desenhos, esquemas ou outros para deixar sua resposta mais clara. Se quiser fazer comentários (para explicar como entendeu uma questão, por exemplo), fique à vontade.

NÃO ESQUEÇA DE IDENTIFICAR A PROVA ("PROVA 1-SEMESTRE 2021-2") E COLOCAR SEU NOME E O NÚMERO DE MATRÍCULA!

BOA PROVA!

Questão 1 – 3 pontos)



Você já viu uma ‘nuvem de palavras’ por aí, né? É uma forma visual de identificar as palavras mais empregadas em determinado contexto. Também é empregado para, visualmente, identificar temas mais relevantes em textos e discursos (políticos, por exemplo). As palavras que aparecem maiores são as mais frequentes, as menores, as menos frequentes. Com base nesse tipo de uso, responda às questões abaixo.

a) Você precisa montar uma tabela com palavra e quantidade de vezes que ela aparece, certo? **Se você fosse um calouro**, iria implementar essa tabela usando uma **lista simples encadeada**. Mostre qual seria o algoritmo básico para armazenar 1 nova palavra numa lista simples (não uma lista de python, mas uma implementada por você mesmo). Considere que existe uma função “LêPróximaPalavra” que devolve a próxima palavra do texto ou None se o texto acabou.

b) Agora, você já está na 3ª fase, isso de listas é coisa de calouros, né? Então você está pensando em um HASH. Como seria a estratégia para colocar essas palavras em um HASH (descreva, não precisa de um algoritmo)? Isso seria melhor que colocar as palavras em uma lista? Como? Por quê?

c) Mas aí aparecem outros requisitos: Qual a palavra mais usada e quantas vezes ela foi utilizada? O HASH responde bem a isso?

d) E uma Árvore Binária de Busca AVL? Serviria para acumular as informações e depois responder à questão de “Qual a mais usada” ?. Se for possível, como você faria? Se não for possível, por que não?

Questão 2 – 2 pontos) Um hash tradicional (com encadeamento direto) é composto por uma tabela de espalhamento em um array de listas (ou seja, o desenho tradicional de hash).

Imagine que, em vez do array, alguém resolve implementar uma lista encadeada para a tabela de espalhamento, na expectativa de flexibilizar a coisa (se precisar aumentar a tabela, seria mais fácil com lista encadeada do que com um array clássico). A pergunta que não quer calar é: “Faz sentido (o cara é um gênio) ou é uma bobagem (é um bobo alegre)”?

- a) Liste quais aspectos são importantes para fazer essa análise (gênio X bobão) ?
- b) Analise esses aspectos importantes fazendo essa comparação de implementações (2 pontos)

Questão 3 – 2 pontos) Ainda sobre Hashs! Pense que você está com um hash bem grande (muuuuitos dados – 1 milhão, tabela de espalhamento grande, fator de carga 4), carinhosamente chamado de Monstrinho. Pois bem, chega uma demanda para SEPARAR os dados em 2 estruturas de Hash separadas, colocando todos os elementos que atendam a um determinado critério em um hash (que chamaremos de Filhote) e os demais ficarão num hash (que será chamado Pequenino).

Idéia: Tenho que passar por cada um dos elementos do Monstrinho para ver se ele atende ao critério e, caso positivo, retirá-lo do Monstrinho e incluí-lo no Filhote.

Analise e comente:

- a) Isso afetaria o fator de carga do Monstrinho? Como / por quê?
- b) Os elementos que não atenderem ao critério podem ser deixados no Monstrinho (e bastaria renomeá-lo para Pequenino) ou precisariam ser também removidos e incluídos em Pequenino? Por quê?
- c) Qual o intervalo possível para o valor do fator de carga do Filhote? Por quê? E em relação à estrutura hash dos elementos que não forem para Filhote (aqueles que ficaram no Monstrinho ou que foram movidos para Pequenino), como ficará o fator de carga? Cresce, diminui, em qual proporção?
- d) Apresente um algoritmo simples que faça a ‘extração’ de Monstrinho dos elementos que atendem ao critério? Será que eu preciso usar a Função de Hash para isso?

Questão 4 – 2 pontos) Considere que tenho 2 árvores binárias de busca, que estão balanceadas, A1 e A2, com elementos de mesmo tipo mas sem elementos repetidos (não se repetem nem na mesma árvore nem nas duas árvores – ou seja, se há um elemento 42 na árvore A1, não haverá um elemento 42 na árvore A2).

Preciso uni-las em uma única árvore binária de busca balanceada. Resolvi usar a seguinte estratégia: *Escolho uma das árvores para começar e pego elementos da outra (um a um) e incluo na primeira.*

Avalie e explique:

- a) Vou conseguir gerar uma árvore binária de busca BALANCEADA com essa estratégia? Por quê?
- b) Devo escolher a árvore para começar que seja a de maior altura ou a de menor altura? Por quê?
- c) Para pegar os elementos da outra, um a um, devo optar por qual caminhada (pré, in ou pós-ordem)? Por quê?
- d) Considerando que a altura de A1 é X e a de A2 é Y, quais poderiam ser as alturas da árvore resultante? Por quê?

Questão 5 – 1 ponto) Avalie e COMENTE/ARGUMENTE/DISCUТА: Os algoritmos de inclusão e de remoção de elementos de uma árvore binária de busca são importantes para definição da organização da árvore? Ou seja, se ela vai ficar balanceada ou não? E a ordem de entrada dos dados? Afeta a organização?