Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Algoritmos e Estrutura de Dados I – CIC102 / 20092
Professor: David Menotti (menottid@gmail.com)

Trabalho Prático 1 - Vetores

Tipos Abstratos de Dados / Alocação Dinâmica

Valor: 0,5 pontos (5% da nota total) **Data de entrega: 15/09/2009**
Documentação em Latex: 0,1 pontos extra
Impressão frente-verso: 0,05 pontos extra

Objetivos

Consiste em rever conceitos básicos de programação e alocação de memória bem como explorar os conceitos de Tipos Abstratos de Dados (TADs). Ainda serão necessários conhecimentos em geometria analítica para realização do trabalho prático.

Descrição

Você deverá implementar um tipo abstrato de dados `TVetor` para representar vetores no espaço $\Re^n$. Esse tipo abstrato deverá armazenar a dimensão do vetor e suas respectivas coordenadas. Considere que a dimensão dos vetores é determinada em tempo de execução (alocação dinâmica).

As operações que devem ser realizadas pelo seu TAD são:

1. Ler um vetor (leia dimensão e coordenadas);
2. Criar um vetor (forneça dimensão e coordenadas, sendo que o número de parâmetros da função varia de acordo com a dimensão do vetor, assemelhando-se à função "`printf`", onde a formatação do texto determina o número de parâmetros);
3. Calcular a soma de dois vetores;
4. Calcular a substração de dois vetores;
5. Calcular o produto escalar entre dois vetores;
6. Calcular o produto vetorial entre dois vetores;
7. Calcular o produto misto entre dois vetores;
8. Imprimir um vetor.

Você deverá implementar também um tipo abastrado de dados `TConj` para armazenar um conjunto de vetores que pode crescer e diminuir durante a execução do programa.

Esse TAD deve implementar as seguintes operações:

1. Inicializar um conjunto como vazio;
2. Somar todos os vetores do conjunto;
3. Adicionar um vetor ao conjunto;
4. Remover um vetor V;
5. Unir dois conjuntos de vetores;
6. Imprimir um conjunto de vetores.

A determinação das declarações das operações (ou assinaturas das funções) faz parte da

avaliação do trabalho e é de inteira responsabilidade sua.

Implemente seus TADs em arquivos separados do programa principal (`TVetor.c` e `TVetor.h e TConj.c e TConj.h`). Se necessário, você pode criar outras funções auxiliares em seu TAD. Suas funções devem executar testes de consistência.

Uma vez criado os TADs, você deverá criar um programa *main* para testá-lo com várias hipóteses de erro.

O que deve ser entregue

- Código fonte do programa em C ou C++ (bem identada e comentada).
- Documentação do trabalho. Entre outras coisas, a documentação deve conter:
  1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados na implementação devem ser inseridos na documentação.
  3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementados e do programa como um todo (notação O).
  4. Listagem de testes executados: os testes executados devem ser simplesmente apresentados.
  5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização
  7. Em Latex: Caso o trabalho seja elaborado/escrito em latex, ganha-se 0,1 pontos.
  8. Formato: mandatoriamente em PDF (http://www.pdf995.com/).

  Obs: Veja modelo de como fazer o trabalho em latex:
  http://www.decom.ufop.br/prof/menotti/aedI092/tps/modelo.zip
  (caso alguém desenvolva um modelo similar em word, favor me enviar)

Como deve ser feita a entrega:

A entrega DEVE ser feita via Moodle (www.decom.ufop.br/moodle) na forma de um **único** arquivo zipado, contendo o código, os arquivos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho.

Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Algoritmos e Estrutura de Dados I – CIC102 / 20092
Professor: David Menotti (menottid@gmail.com)

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, identação e comentários no programa também vão valer pontos;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero;
- Trabalhos entregue em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.