



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA MINAS GERAIS
CAMPUS FORMIGA

Trabalho Prático

Professor: *Felipe Reis*

Disciplina: *Matemática Discreta*
Data de entrega: 17/01/2022

Objetivos

- Aprofundar o conhecimento em conceitos de Conjuntos e Relações, ensinados em sala de aula;
- Estimular o formalismo matemático e a programação de software.

Recomendações

Leia atentamente todos os requisitos deste documento.

Os pontos serão distribuídos de acordo com o cumprimento das exigências definidas em cada etapa. Não faça atividades extras: elas não serão valorizadas. Dê maior atenção aos requisitos cuja pontuação é mais alta.

Todas as decisões tomadas para atender aos requisitos deste documento devem ser baseadas em teorias de Conjuntos e Relações.

Requisitos

O trabalho será dividido em 4 exercícios:

1. Implementação de operações de conjuntos;
2. Análise de subconjuntos;
3. Avaliação de propriedades de relações;
4. Implementação de fechos reflexivos e simétricos.

Para cada uma das questões devem ser gerados algoritmos implementados em linguagem C/C++.

Exercício 1.

Considere como universo de valores o conjunto de números inteiros $0 \leq x \leq 50$. Considere, ainda, dois conjuntos A e B , contendo 10 elementos aleatório cada, dentro do universo de valores possíveis.

Devem ser criados algoritmos proceder as seguintes operações:

- $A \cup B$
- $A \cap B$
- $A - B$
- $B - A$
- $\neg A$

O sistema deverá imprimir os conjuntos originais, A e B , e, em seguida, imprimir o resultado das operações. Podem ser criadas funções para implementação de cada uma das operações, evitando repetição de trabalho.

Recomendação 1: Utilize sementes fixas para gerar números aleatórios durante o desenvolvimento, de modo a facilitar a conferência de valores. Após a construção do algoritmo, altere o código para adição de sementes aleatórias.

Recomendação 2: Ordene os conjuntos para facilitar a implementação e aumentar o desempenho do algoritmo.

Exercício 2.

Considere como universo de valores o conjunto de números inteiros $0 \leq x \leq 50$.

Considere dois conjuntos A e B , onde $|A| = 20$ e $|B| = 5$, compostos por elementos aleatórios dentro do universo de valores possíveis.

Deve ser criado um algoritmo para analisar se $B \subseteq A$.

Recomendação 1: Utilize sementes fixas para gerar números aleatórios durante o desenvolvimento, de modo a facilitar a conferência de valores. Após a construção do algoritmo, altere o código para adição de sementes aleatórias.

Recomendação 2: Ordene os conjuntos para facilitar a implementação e aumentar o desempenho do algoritmo.

Exercício 3.

Considere como universo de valores o conjunto de números inteiros $0 \leq x \leq 4$. Gere uma relação aleatória R contendo 10 elementos ordenados (ex.: $R = \{(0,0), (0,4), (1,1), (1,2), (2,0), \dots, (4,1)\}$).

A relação pode ser representada por uma matriz 10×2 , onde cada linha representa uma tupla.

Análise se a relação gerada aleatoriamente é:

- Reflexiva;
- Simétrica;
- Antissimétrica;
- Transitiva.

Recomendação: Utilize sementes fixas para gerar números aleatórios durante o desenvolvimento, de modo a facilitar a conferência de valores. Após a construção do algoritmo, altere o código para adição de sementes aleatórias.

Exercício 4.

Considere como universo de valores o conjunto de números inteiros $0 \leq x \leq 4$. Gere uma relação aleatória R contendo 10 elementos ordenados (ex.: $R = \{(0,0), (0,4), (1,1), (1,2), (2,0), \dots, (4,1)\}$).

A relação pode ser representada por uma matriz 10×2 , onde cada linha representa uma tupla.

Construa um algoritmo para indicar os seguintes fechos:

- Reflexivo;
- Simétrico.

Recomendação 1: Utilize sementes fixas para gerar números aleatórios durante o desenvolvimento, de modo a facilitar a conferência de valores. Após a construção do algoritmo, altere o código para adição de sementes aleatórias.

Recomendação 2: Crie um array de 20 elementos para indicar os fechos reflexivos e simétricos. Apesar de não ser a melhor forma de implementação, devido ao desperdício de memória, ela poderá ser adotada. No entanto, não imprima registros extras (em branco, null, etc).

Entrega do Trabalho.

Os trabalhos deverão ser entregues em arquivos `.c`, `.cpp` ou `.ipynb`, contendo o número do exercício (ex.: `exercicio1.cpp`, `exercicio2.c`, etc).

Deve ser entregue, ainda, um arquivo README, em formato texto (`.txt` ou `.md`), contendo informações acerca do trabalho, ferramentas utilizadas e ferramenta de compilação.

Os algoritmos deverão ser compilados utilizando os compiladores GCC, G++ ou Xeus CLing.

Ponto Extra: Trabalhos publicados em repositórios públicos open-source (ex.: Github, Bitbucket, GitLab, SourceForge, etc) serão valorizados em 1 ponto extra (enviar link do repositório).

Data de Entrega, Organização de Grupos e Pontuação

O trabalho deve ser entregue até **17/01/2022**, na atividade no Google Classroom.

O trabalho possui valor de **20 pontos**.

O trabalho poderá ser realizado em grupos de **até 2 alunos**.

Critérios de avaliação

A tabela abaixo contém os critérios de avaliação e a distribuição dos pontos dentre as tarefas do trabalho.

Exercício	Descrição	Pontuação
1	Implementação das operações União, Interseção e Complemento (1,75 pts cada).	7 pontos
	Implementação das operações de Diferença (1,25 ponto cada).	
	Qualidade do código fonte (0,5 pontos).	
2	Análise de subconjuntos (1,5 pontos).	2 pontos
	Qualidade do código fonte (0,5 pontos).	
3	Análise das propriedades reflexiva, simétrica e antissimétrica (1,5 pts cada).	7,5 pontos
	Análise das propriedades transitiva (2,5 pontos).	
	Qualidade do código fonte (0,5 pontos).	
4	Implementação dos fechos reflexivos e simétricos (1,5 pts cada).	3,5 pontos
	Qualidade do código fonte (0,5 pontos).	

Atenção: Algoritmos cujos conjuntos forem gerados sem utilização de números aleatórios serão desvalorizados em no mínimo 50%.