

# Algoritmos e Estruturas de Dados I

## Projeto I I: Operações sobre conjuntos (Sets)

Professores Marcelo G. Manzato e Rudinei Goularte  
(mmanzato@icmc.usp.br, rudinei@icmc.usp.br)

Data de Entrega: 08/12/2022

Este projeto deverá ser feito em grupos de 3 alunos. A entrega deve ser realizada no Tidia (veja abaixo).

### Descrição do Problema

**Conjunto** é um conceito básico do ramo matemático da Teoria dos Conjuntos. Um Conjunto representa uma coleção de objetos, em que cada objeto é denominado **elemento**. A relação básica entre um elemento e o conjunto é a relação de pertinência, que determina se um elemento pertence ou não ao conjunto.

Conjuntos tem diversas aplicações em sistemas computacionais, estando presentes em várias linguagens de programação (como uma ED), assim como em soluções de problemas matemáticos (puros e aplicados), de otimização e estatísticos.

### Objetivo do Trabalho

Implementar o TAD Conjunto de modo que a Estrutura de Dados escolhida possibilite que as operações **indicadas** tenham **a menor complexidade computacional possível**. Os elementos de cada conjunto são representados por Itens (TAD Item). As operações consideradas são:

- 1) **Operações básicas** para TADs em geral: Criar (um conjunto), Apagar (um conjunto), **Inserir** (um elemento em um conjunto) –  $O(\log n)$ , **Remover** (um elemento de um conjunto) –  $O(\log n)$  e Imprimir (imprimir os elementos/itens armazenados no TAD).

- 2) As **operações específicas** de conjuntos: **Pertence** (um elemento está presente ou não no conjunto) –  $O(\log n)$ , **União** –  $O(n \log n)$  e **Intersecção** –  $O(n \log n)$ . Note que essas 2 últimas operações recebem como entrada 2 conjuntos e devolvem um terceiro conjunto como resposta.

### Entrega do Trabalho

1) Arquivo TXT: um breve relatório justificando a(s) Estrutura(s) de Dados escolhida(s) para representar Conjuntos e a complexidade computacional de tempo (Big Oh) de cada uma das operações (básicas e específicas).

2) Arquivo ZIP: Crie um arquivo Nome\_Grupo.zip (Nome\_Grupo deve ser substituído pelo nome do grupo) contendo: o arquivo TXT (relatório), o makefile do projeto e **todo código fonte** necessário para compilar e executar o programa, incluindo arquivos *.c* (*main* e outros), arquivos *.h* e *arquivos de teste*. **Não adicionar arquivos executáveis nem arquivos objeto.**

3) O arquivo .ZIP deve ser entregue no Tidia (Escaninho) até 08/12/2022, 23:59h. Trabalhos submetidos em datas posteriores à especificada terão 1,0 ponto de desconto por dia de atraso. Apenas um integrante do grupo precisa submeter o trabalho.

### Observações Importantes

- O projeto deve ser desenvolvido em grupos de 3 alunos.
- A implementação deve ser realizada em linguagem C, padrão C99.
- Os Itens, por simplicidade e conveniência, armazenam uma única informação (chave) do tipo **int**. Porém, deve-se lembrar que formalmente Itens podem armazenar qualquer arranjo de informações e, assim, não se podem fazer suposições ou simplificações sobre seu conteúdo, ordem (ordenação), volume e etc. Por exemplo, a solução para os Conjuntos deveria funcionar do mesmo modo se em vez de **int** cada Item armazenasse um **char**, um **float** ou um vetor de **structs**.
- Não serão fornecidos arquivos de teste. Cada grupo será responsável por desenvolver seus próprios testes e testar seu projeto antes da entrega. **Dica 1:** façam arquivos csv/txt com entradas variadas e usem os “operadores” < e > no shell para carregar/criar os arquivos de testes. **Dica 2:** façam testes (mais de um!) para todas as operações. Um mesmo arquivo pode testar uma combinação de operações (criação+inserções+imprimir+apagar, p.e). Atenção especial aos testes das operações específicas. **Dica 3:** os PAEs podem ajudar, ensinando a gerar os casos de teste. Consultem-nos!!!
- Os projetos serão compilados e corrigidos utilizando o compilador GCC Linux.
- Haverá, obviamente, verificação de plágio. Aos projetos envolvidos em plágio será atribuída nota zero.

### Critérios de Avaliação

- Aderência à especificação e ao uso de TADs (4 pontos).

- Eficácia e eficiência da solução (4 pontos).
- Documentação interna (1 ponto).
- Relatório (1 ponto).

### **Exemplo de programa principal e interface (cliente/main.c e set.h)**

- Consultem o Tidia: Repositório/Códigos/

Arquivos Exemplo\_main\_projeto2.c e set.h

### **Exemplos de entradas e saídas:**

Entrada 1:

```
5
3
1 2 3 4 5
7 2 8
2
```

Saída 1:

Pertence.

Entrada 2:

```
5
3
1 2 3 4 5
7 2 8
2
```

Saída 2:

1, 2, 3, 4, 5, 7, 8,

**Bom Trabalho!**