

Atividade Acadêmica de Estrutura de Dados 1



Aluno: Matheus Felipe Correa Alves

Matrícula: 2016780361

Professor: Marcelo Zamith

Período: 2017.2

Resumo da Tarefa

Implementar um programa que permita criar e manipular conjuntos, conforme as opções: *criar conjunto*, *destruir conjunto*, *pesquisar um elemento em um conjunto*, *união de dois conjuntos* e, *intersecção de dois conjuntos*.

Defini-se conjunto uma árvore que está contida em uma lista. A lista é do tipo simplesmente encadeada e possui todos os conjuntos.

Entrega

Programa implementado em C. Possui os seguintes arquivos: *Main.c* *Arvore.h* *Arvore.c* *Lista.h* *Lista.c*

Linha para compilar: `gcc -o teste Main.c Arvore.c Lista.c`

Link do repositório no github: https://github.com/Felcks/estrutura_de_dados

Estruturas Principais

Lista.h

```
typedef struct elementoLista {  
    unsigned int indice;  
    struct elementoLista *prox;  
    Arvore *arvore;  
} ElementoLista;  
  
typedef ElementoLista* Lista;
```

Arvore.h

```
typedef struct elemento {  
    unsigned int indice;  
    char site[256];  
    struct elemento *dir;  
    struct elemento *esq;  
    struct elemento *pai;  
} Elemento;  
  
typedef Elemento* Arvore;
```

O **ElementoLista** possui um índice, um ponteiro para o próximo elemento e um ponteiro para o primeiro elemento da árvore.

Um **Elemento** possui um índice, um site e ponteiros para o elemento da direita, esquerda e o pai.

Das Funções do Programa

- 1 - Criar conjunto.
- 2 - Destruir conjunto.
- 3 - Pesquisar um elemento em um conjunto.
- 4 - União de dois conjuntos.
- 5 - Intersecção de dois conjuntos.
- 6 - Exibir árvores.
- 7 - Exibir todos os elementos de um conjunto em ordem.
- 8 - Exibir altura da árvore/árvore balanceada.
- F - Fim do programa.

*As funções de 1, 2, 3, 6, 7, 8 são triviais e não serão abordadas nesse artigo.

4 - União de dois conjuntos.

A operação de união consiste em criar um novo conjunto com todos os elementos do conjunto "A" e do conjunto "B". Onde o programa vai considerar o índice dos elementos para eliminar as repetições ou os elementos de interseção. O programa deverá permitir a união de dois conjuntos vazios.

A função de copiar uma árvore para outra passa, ao mesmo tempo, por todos os elementos da árvore a ser copiada e a árvore a receber a cópia em percurso pré-ordem. Ela adiciona um elemento somente quando a árvore a receber a cópia não possui um elemento naquela posição

Supondo que nós temos duas árvores nas variáveis `arvore_A` e `arvore_B`. A união dessas duas árvores foi feita da seguinte forma:

- 1) Criamos uma nova árvore, chamada `arvore_C`;
- 2) Copiamos todos os índices de `arvore_A` para `arvore_C`.
- 3) Copiamos todos os índices de `arvore_B` para `arvore_C` com excessões das repetições.

5 - Intersecção de dois conjuntos.

A operação de intersecção consistem em criar um novo conjunto com os elementos que estão no conjunto "A" e também estão no conjunto "B". Onde o programa vai considerar o índice dos elementos. Na operação de interseção, o programa deverá permitir a criação de conjunto vazio.

Supondo que nós temos duas árvores nas variáveis `arvore_A` e `arvore_B`. A intersecção dessas duas árvores foi feita da seguinte forma:

- 1) Criamos uma nova árvore, chamada `arvore_C`;
- 2) Percorremos a `arvore_A` e a `arvore_B` ao simultaneamente.
 - a) Se o elemento não for nulo e compartilharem de mesmo índice em ambas as árvores, então adicionamos esse elemento na árvore a receber a intersecção.