

Estrutura de Dados: Conjuntos Dinâmicos

Atílio G. Luiz

25 de abril de 2023

1 Introdução

Conjuntos são tão fundamentais para a Ciência da Computação quanto para a Matemática. Enquanto os conjuntos matemáticos são invariáveis, os conjuntos manipulados por algoritmos podem crescer, encolher ou sofrer outras mudanças ao longo do tempo. Chamamos tais conjuntos de **conjuntos dinâmicos**.

Algoritmos podem exigir a execução de vários tipos diferentes de operações em conjuntos. Por exemplo, muitos algoritmos precisam apenas da capacidade de inserir e eliminar elementos em um conjunto e testar a pertinência de elementos em um conjunto. Damos o nome de **dicionário** ao conjunto dinâmico que suporta essas operações. Outros algoritmos exigem operações mais complicadas. A melhor maneira de implementar um conjunto dinâmico depende das operações que devem ser suportadas.

1.1 Elementos de um conjunto dinâmico

Em uma implementação típica de um conjunto dinâmico, cada elemento é representado por um objeto cujos atributos podem ser examinados e manipulados se tivermos um ponteiro para o objeto. Alguns tipos de conjuntos dinâmicos consideram que um dos atributos do objeto é uma **chave** de identificação. Se as chaves são todas diferentes, podemos imaginar o conjunto dinâmico como um conjunto de valores de chaves. O objeto pode conter **dados satélites**, que são transportados em atributos de outro objeto mas que, fora isso, não são utilizados pela implementação do conjunto. Também pode ter atributos que são manipulados pelas operações de conjuntos; esses atributos podem conter dados ou ponteiros para outros objetos no conjunto.

Alguns conjuntos dinâmicos pressupõem que as chaves são extraídas de um conjunto totalmente ordenado como o dos números reais ou o de todas as palavras sob a ordenação alfabética usual. Uma ordenação total nos permite definir o elemento mínimo do conjunto, por exemplo, ou falar do próximo elemento maior que um dado elemento em um conjunto.

1.2 Operações em Conjuntos Dinâmicos

As operações em um conjunto dinâmico podem ser agrupadas em duas categorias: **consultas**, que simplesmente retornam informações sobre o conjunto, e **operações modificadoras**, que alteram o conjunto. Apresentamos a seguir, uma lista de operações típicas.

- **Search(S, k)**

Uma consulta que, dado um conjunto S e um valor de chave k , retorna um ponteiro x para um elemento em S tal que $x.chave = k$ ou NIL se nenhum elemento desse tipo pertencer a S .

- **Insert(S, x)**

Uma operação modificadora que aumenta o conjunto S com o elemento apontado por x . Normalmente, consideramos que quaisquer atributos no elemento x necessários para a implementação do conjunto já foram inicializados.

- **Remove(S, x)**

Uma operação modificadora que, dado um ponteiro x para um elemento no conjunto S , remove x de S . (Observe que essa operação utiliza um ponteiro para um elemento x , não um valor de chave.)

- **Minimum(S)**
Uma consulta em um conjunto totalmente ordenado S que retorna um ponteiro para o elemento de S que tenha a menor chave.
- **Maximum(S)**
Uma consulta em um conjunto totalmente ordenado S que retorna um ponteiro para o elemento de S que tenha a maior chave.
- **Successor(S, x)**
Uma consulta que, dado um elemento x cuja chave é de um conjunto totalmente ordenado S , retorna um ponteiro para o elemento maior seguinte em S ou NIL se x é o elemento máximo.
- **Predecessor(S, x)**
Uma consulta que, dado um elemento x , cuja chave é de um conjunto totalmente ordenado S , retorna um ponteiro para o elemento menor seguinte em S ou NIL se x é o elemento mínimo.

Em geral, medimos o tempo empregado para executar uma operação de conjunto em termos do tamanho do conjunto. Por exemplo, vimos a estrutura de dados Árvore AVL que pode suportar qualquer das operações da lista apresentada acima em um conjunto de tamanho n no tempo $O(\lg n)$.

2 Diagrama de classes

Vamos representar um conjunto dinâmico por meio de uma classe base abstrata chamada **dynamic_set**. Por ser uma classe abstrata, ela servirá apenas para polimorfismo, dado que não será possível instanciar nenhum objeto do tipo **dynamic_set**. A classe **dynamic_set** terá como classes derivadas a classe **avl_tree** e a classe **rb_tree**. Essa hierarquia está ilustrada na Figura 1.

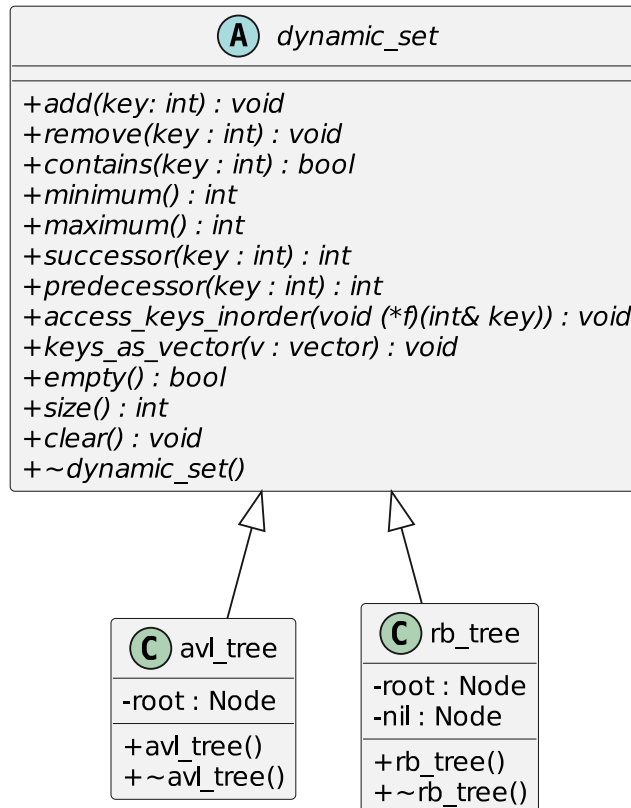


Figura 1: Diagrama de classes da hierarquia que compõe a estrutura de dados conjunto dinâmico.

3 Definição da classe abstrata em C++

```
1 // file dynamic_set.h
2 #ifndef DYNAMIC_SET_H
3 #define DYNAMIC_SET_H
4 #include <vector>
5
6 /**
7  * Abstract class 'dynamic_set'.
8  * This class declares pure virtual functions that
9  * express the basic functionalities of a dynamic set data structure.
10 * A dynamic set is a set that can increase and decrease in size.
11 */
12 class dynamic_set {
13 public:
14     virtual void add(int key) = 0; // 0(lg n)
15     virtual void remove(int key) = 0; // 0(lg n)
16     virtual bool contains(int key) const = 0; // 0(lg n)
17     virtual int minimum() const = 0; // 0(lg n)
18     virtual int maximum() const = 0; // 0(lg n)
19     virtual int successor(int key) const = 0; // 0(lg n)
20     virtual int predecessor(int key) const = 0; // 0(lg n)
21     virtual void access_keys_inorder(void (*f)(int& key)) = 0; // 0(n)
22     virtual void keys_as_vector(std::vector<int>& v) const = 0; // 0(n)
23     virtual bool empty() const = 0; // 0(1)
24     virtual int size() const = 0; // 0(n)
25     virtual void clear() = 0; // 0(n)
26     virtual ~dynamic_set() = default;
27 };
28
29 #endif
```

dynamic_set.h

Referências

- [1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2001). Introduction to Algorithms. The MIT Press. ISBN: 0262032937