Estrutura de Dados I

Prof. Ms. *Rogério Ad. Sousa*

Deque e Set

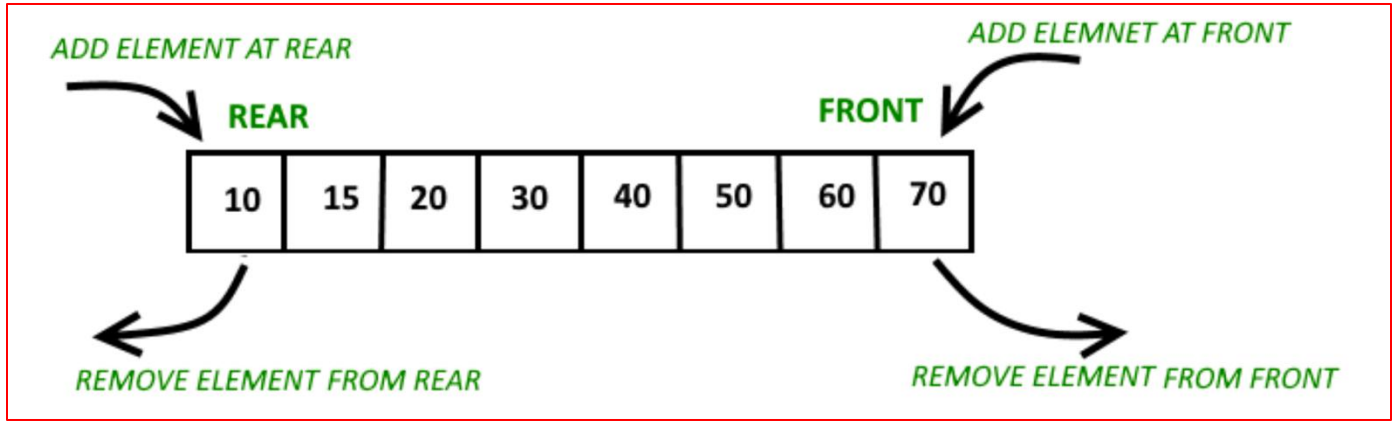
- Definição
- Operações
- Aplicações
- Funcionalidades
- SET
- Utilizaçao

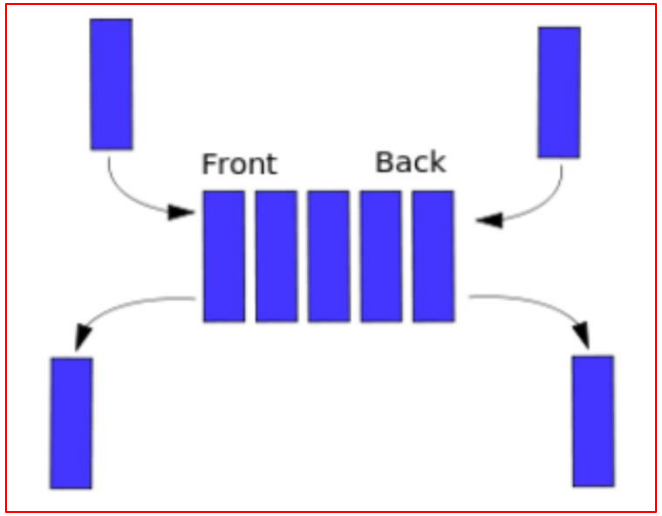
Definição

- Um deque (double-ended queue) é uma estrutura de dados que combina as características de uma pilha (stack) e uma fila (queue).
- Um deque permite inserções e remoções eficientes tanto no início quanto no final da estrutura.
- A palavra "deque" é pronunciada como "deck", e é uma abreviação de "double-ended queue" (fila de duas extremidades).

DEQUES - DOUBLE-ENDED QUEUE (Operações)

- Um deque permite uma variedade de operações, como empilhar e desempilhar elementos em ambas as pontas, adicionar ou remover elementos da frente ou do final da estrutura e percorrer a fila em ambas as direções.





DEQUES - DOUBLE-ENDED QUEUE (Aplicação)

- Uma das aplicações mais simples do Deque ao mundo real seria a de filas com acesso prioritário (bancos, hospitais, etc..) onde a maioria das entidades segue a lógica padrão de fila, mas existem casos prioritários que serão empurrados para a frente desta fila.

DEQUES - DOUBLE-ENDED QUEUE (Funcionalidades)

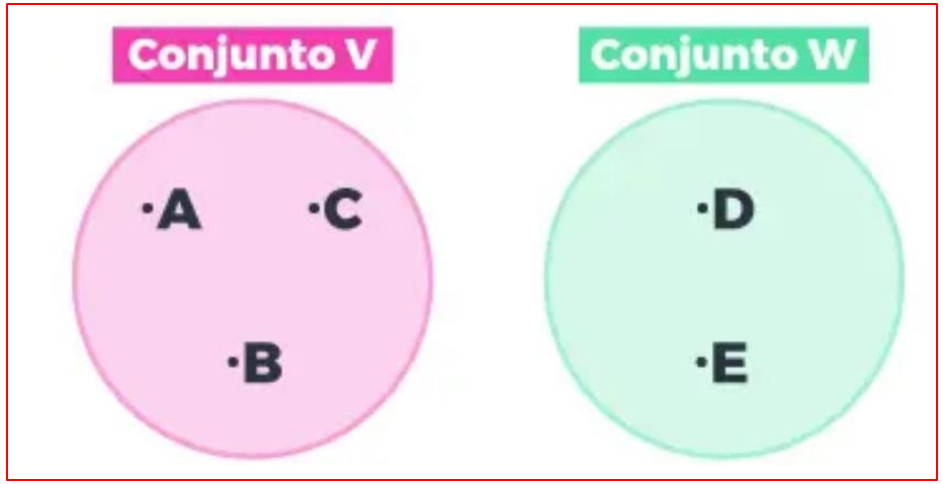
- push_front(valor) - adiciona um elemento no início da deque;
- push_back(valor) - adiciona um elemento no final da deque;
- pop_front() - remove o elemento no início da deque;
- pop_back() - remove o elemento no final da deque;
- size() - retorna o número de elementos na deque;
- empty() - retorna true se a deque estiver vazia;
- front() - retorna o primeiro elemento da deque;
- back() - retorna o último elemento da deque;
- at(index) - retorna o elemento na posição especificada;
- clear() - remove todos os elementos da deque;

Em resumo...

- Um deque, também conhecido como Double-ended Queue, é uma estrutura de dados que permite a inserção e remoção de elementos tanto no início quanto no final da fila.
- Nesta implementação, o deque é baseado em uma lista encadeada duplamente, em que cada elemento é representado por um nó contendo referências para o nó anterior e próximo.

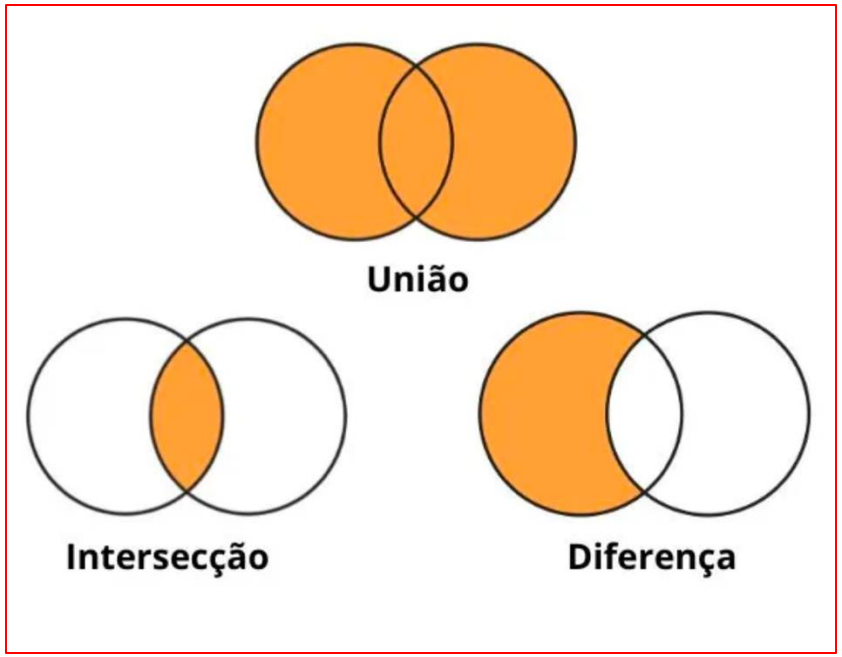
SET - CONJUNTO (Definição)

- SET é uma estrutura de dados que armazena um conjunto de valores únicos.
- Em um conjunto, não é possível repetir valores, não importa a ordem dos elementos.
- A maior parte das linguagens de programação mais usadas têm métodos nativos para criação de conjuntos.



SET - CONJUNTO (Uso)

- A estrutura do conjunto vem da matemática, e também é possível fazer operações como união e intersecção em conjuntos de dados.
- Um dos usos mais comuns desta estrutura é em bancos de dados SQL.

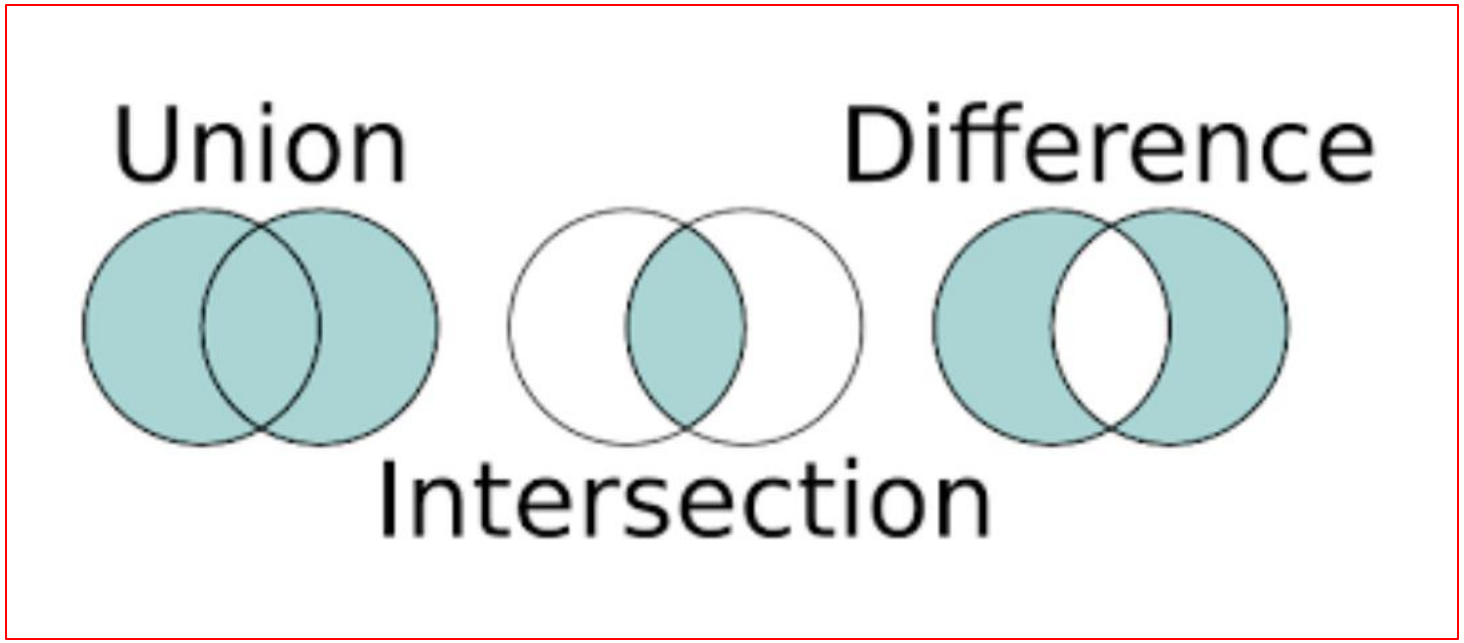


SET - CONJUNTO (Funcionalidades)

- insert(): insere um elemento no conjunto
- erase(): remove um elemento do conjunto
- clear(): remove todos os elementos do conjunto
- size(): retorna o número de elementos no conjunto
- empty(): retorna true se o conjunto estiver vazio
- find(): procura um elemento no conjunto e retorna um iterador para ele, ou um iterador para o final do conjunto se o elemento não for encontrado

SET - CONJUNTO

- union(): insere um elemento no conjunto
- intersection(): remove um elemento do conjunto
- difference(): remove todos os elementos do conjunto



Deque - Utilização (Exemplo)

- O deque pode ser utilizado da seguinte forma:

1- Crie uma instância do deque:

Deque<T> deque;

Substitua T pelo tipo de dado que deseja armazenar no deque.

Deque - Utilização (Exemplo)

2. Realiza as operações desejadas no deque, como inserção e remoção:

- deque.isEmpty();
- deque.pushFront(value);
- deque.pushBack(value);
- deque.popFront();
- deque.popBack();
- deque.getFront();
- deque.getBack();
- deque.clear();
- deque.print();

Substitua value pelo valor a ser inserido ou removido.

Class DEQUE

- Anexo...

Class SET

- Anexo...

Deque - Prática...

```cpp
#include <iostream>

template <typename T>
class Deque {
    // Código da classe Deque
};

int main() {
    Deque<int> deque;
    deque.pushFront(3);
    deque.pushFront(2);
    deque.pushFront(1);

    std::cout << "Deque: ";
    deque.print();

    deque.pushBack(4);
    deque.pushBack(5);
    deque.pushBack(6);

    std::cout << "Deque: ";
    deque.print();

    std::cout << "Frente: " << deque.getFront() << std::endl;
    std::cout << "Trás: " << deque.getBack() << std::endl;

    deque.popFront();
    deque.popBack();

    std::cout << "Deque após popFront() e popBack(): ";
    deque.print();

    deque.clear();

    std::cout << "Deque após clear(): ";
    deque.print();

    return 0;
}
```

- ✓ Neste exemplo, uma instância da classe Deque é criada para armazenar valores do tipo int.
- ✓ Valores são adicionados no início da deque usando o método pushFront(), e em seguida, são adicionados no final da deque usando o método pushBack().
- ✓ A deque é impressa usando o método print().
- ✓ O elemento da frente da deque é obtido usando o método getFront() e o elemento de trás da deque é obtido usando o método getBack(), e ambos são exibidos na saída padrão.
- ✓ Em seguida, elementos são removidos tanto no início quanto no final da deque usando os métodos popFront() e popBack(), respectivamente.
- ✓ Adeque é impressa novamente para verificar as mudanças.
- ✓ Por fim, a deque é limpa usando o método clear() e novamente impressa, mostrando que está vazia.

SET - Prática...

```cpp
int main() {
    Set<int> set;

    set.insert(1);
    set.insert(2);
    set.insert(3);

    std::cout << "Conjunto: ";
    set.print();

    std::cout << "Conjunto contém 2: " << (set.contains(2) ? "true" : "false") << std::endl;
    std::cout << "Conjunto contém 4: " << (set.contains(4) ? "true" : "false") << std::endl;

    set.remove(2);

    std::cout << "Conjunto após remover o 2: ";
    set.print();

    set.clear();

    std::cout << "Conjunto após limpar: ";
    set.print();

return 0;
}
```

Atividades

- Crie um programa que usa um deque para armazenar e exibir uma sequência de números inteiros. Adicione e remova elementos do início e do final do deque e exiba o conteúdo após cada operação.
- Crie um programa que utiliza iteradores para acessar e modificar elementos em um deque. Inicialize o deque com alguns valores, modifique-os usando iteradores e exiba o resultado.
- Implemente uma função que recebe um deque e um valor inteiro. A função deve adicionar o valor ao final do deque, remover o primeiro elemento, e retornar o deque atualizado. Teste a função no main.
- Crie um programa que usa um set para armazenar números inteiros e exibe os valores em ordem crescente. Adicione e remova elementos do set e exiba o conteúdo após cada operação.

Atividades

- Crie um programa que usa um set e inclui funções para verificar se um determinado valor está presente no set e para encontrar o valor mais próximo de um dado valor.
- Crie um programa que usa um set para armazenar números inteiros e conte quantos elementos estão presentes no set. Adicione e remova elementos e exiba a quantidade atual de elementos.
- Crie um programa que utiliza um deque de deque<int> para representar uma matriz 3x3. Preencha a matriz com valores fornecidos pelo usuário e exiba a matriz.
- Crie uma classe Pessoa com atributos nome e idade. Utilize um set para armazenar objetos dessa classe, e defina um critério de comparação para que o set funcione corretamente.