

Desafios de Programação

Prof. Eduardo Theodoro

Universidade Federal de Mato Grosso do Sul (UFMS)

C++

Onde pesquisar?

<http://www.cplusplus.com/>

STL - Containers (Estruturas de Armazenamento de Dados)

Correspondem às coleções de elementos de um determinado tipo, na forma de gabaritos de classe. Os containers definidos pela STL são:

- ▶ vector - elementos organizados na forma de um array que pode crescer dinamicamente.
- ▶ stack - implementação de uma pilha em C++.
- ▶ queue - implementação de um fila em C++.
- ▶ list - elementos organizados na forma de uma lista duplamente encadeada.
- ▶ map - cada elemento é um par $\langle \textit{chave}, \textit{elemento} \rangle$ sendo que a chave é usada para ordenação da coleção.
- ▶ set - coleção ordenada na qual os próprios elementos são utilizados como chaves para ordenação da coleção.

Vector

Motivação

Como realizar o cadastro de clientes em um software sem saber previamente quantos clientes você tem?



Vector

Declaração

```
vector< tipo > v;
```

onde *v* é o nome da variável que você escolher.

Insere um elemento na última posição do vetor `v. push_back()`

Acessa um elemento do vetor `v[]`

Apaga um elemento do vetor `v.erase()`

Retorna o tamanho do vetor `v.size()`

Insere um elemento em qualquer posição do vetor `v.insert()`

Apaga todo o vetor `v.clear()`

Troca o conteúdo de dois vetores `v.swap()`

Curiosidade

Como trocar o valor de duas variáveis sem utilizar uma variável auxiliar?

Modo 1

$$a = a + b$$

$$b = a - b$$

$$a = a - b$$

Modo 2 - *xor Trick*

$$a = a \wedge b$$

$$b = a \wedge b$$

$$a = a \wedge b$$

Stack (Pilha)

Ideia

Elementos são sempre inseridos/removidos apenas do final (topo) da pilha.

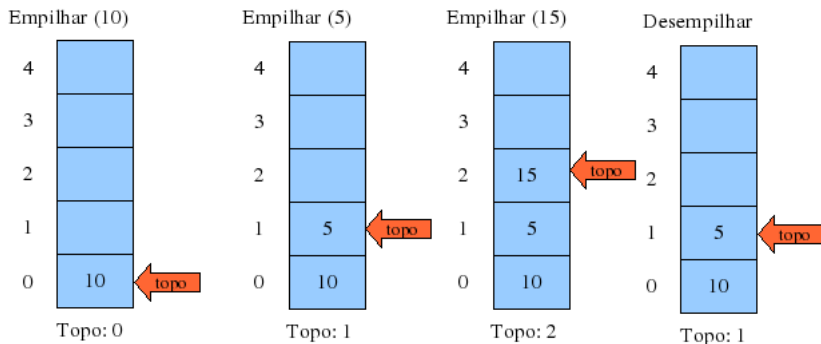


Figura 1 : Exemplo do funcionamento de uma pilha.

Stack

`stack< tipo > s;`

onde *s* é o nome da variável que você escolher.

Inserir um elemento no topo da pilha `s.push()`

Acessar o elemento do topo da pilha `s.top()`

Remover o elemento do topo da pilha `s.pop()`

Retorna o tamanho da pilha `s.size()`

Exercícios

Exercícios sobre Vector e Stack.

A2 Online Judge

UFMS - CPPP - Aula 2. (<https://a2oj.com/edit?ID=29017>)

Queue (Fila)

Ideia

Elementos são sempre inseridos no fim e removidos apenas do início da fila.

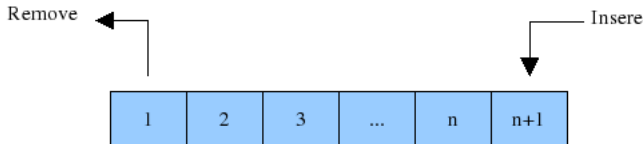


Figura 2 : Exemplo do funcionamento de uma fila.

Queue

`queue< tipo > f;`

onde *f* é o nome da variável que você escolher.

Inserir um elemento no fim da fila `f.push()`

Acessar o elemento do fim da fila `f.back()`

Acessar o elemento do início da fila `f.front()`

Remover o elemento do início da fila `f.pop()`

Retorna o tamanho da fila `f.size()`

Set

Ideia

Estrutura de conjuntos, armazena apenas valores não repetidos. Caso o elemento já exista no conjunto, ele não será inserido. Os elementos de um set estão sempre ordenados.

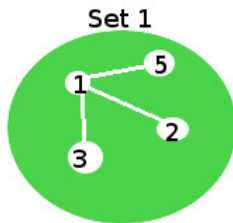


Figura 3 : Exemplo do funcionamento de um set.

Set

`set< tipo > s;`

onde *s* é o nome da variável que você escolher.

Inserir um elemento no conjunto `f.insert()`

Remover um elemento do conjunto `f.erase()`

Retorna o tamanho do conjunto `f.size()`

Set

Acessando um set

Deve-se utilizar iteradores!

Exemplo

```
for(set<int>::iterator it = s.begin(); it!=s.end(); it++)  
    cout<<*it<<endl;
```

Map

Ideia

Implementa a ideia de dicionários. Existe uma chave, que usamos para a busca, e um valor associado a esta chave. As chaves são únicas e estão ordenadas na estrutura.

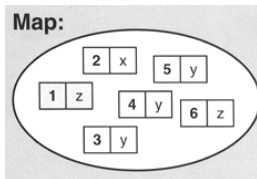


Figura 4 : Exemplo do funcionamento de um map.

Map

`set< tipo > s;`

onde *s* é o nome da variável que você escolher.

Inserir uma tupla no mapa `m.insert()`

Remover um elemento do conjunto `m.erase()`

Retorna o tamanho do conjunto `m.size()`

Busca uma chave no mapa `m.find()`

Map

Acessando um Map

Deve-se utilizar iteradores!

Exemplo

```
for(set<int>::iterator it = m.begin(); it!=m.end(); it++)  
    cout<<(*it).first<<(*it).second<<endl;
```