Estruturas de Dados: Pilhas

Victor Turrisi

Introdução

- Muitas vezes, queremos representar dados do mundo real em um programa de computador

- Dependendo do tipo de dado e de como lidamos com ele, representamos utilizando alguma estrutura de dados

- Por exemplo, podemos representar um mapa por meio de uma matriz, onde cada posição da matriz corresponde à uma coordenada de latitude e longitude do nosso mapa

Introdução

- Para cada tipo de dado, geralmente existe uma estrutura que melhor o representa e que torna mais fácil a sua manipulação

- Nessa aula, veremos a estrutura de dados chamada de Pilha (*stack*)

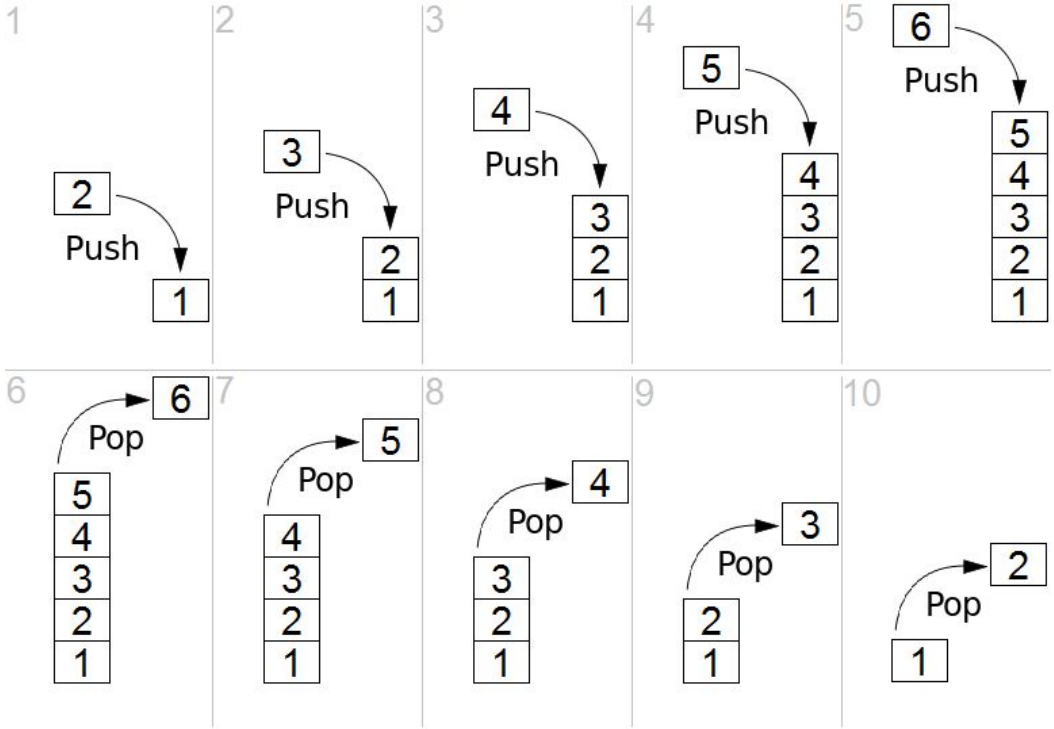
Estrutura de Dado: Pilha

- Uma pilha é um tipo de estrutura de dados no formato LIFO (Last in First out)
- Isso significa que o último elemento que é adicionado a ela, é o primeiro elemento que deve ser removido
- A pilha consiste em um conjunto de elementos que são *empilhados* de acordo com a sua ordem de inserção
- Em quais situações devemos utilizar uma Pilha?

Estrutura de Dado: Pilha

- Ela é composta por duas operações principais: *push* e *pop*
- A operação *push* adiciona um elemento no topo da pilha
- Enquanto a operação *pop* remove o elemento do topo da pilha
- Podemos ainda ter uma operação que mostra o elemento que se encontra no topo da pilha (geralmente chamada de *top*)

Estrutura de Dado: Pilha



Estrutura de Dado: Pilha

- Para representar uma Pilha em C, geralmente utilizamos duas *structs*:
  - Uma é chamada de *Element* (elemento)
  - Enquanto a outra é chamada de *Stack* (pilha)

- A estrutura *Element* irá representar um elemento que existe na pilha

- Por exemplo:

```c
struct Element
{
    int x;
    struct Element *next;
};
```

Estrutura de Dado: Pilha

- Já a struct *Stack* consiste na própria pilha, e possui uma referência para o topo da pilha

```
struct Stack
{
    int size;
    struct Element *top;
};
```

Estrutura de Dado: Pilha

- A partir dessas duas *structs* básicas é possível implementar as funções de uma pilha:

- void push(Stack s, Element e)

- Element pop(Stack s)

- Element top(Stack s)

- int is_empty(Stack s)

Estrutura de Dado: Pilha

- Devemos lembrar que todos os elementos adicionados na Pilha serão alocados dinamicamente e, portanto, ao fim do programa, toda a memória deverá ser desalocada de forma correta

- Isso pode ser feito chamando a operação *pop* enquanto a pilha ainda tiver elementos

Trabalho

- Implemente uma estrutura de Pilha criando as duas *structs* e as três funcionalidades apresentadas

- Cada elemento da sua pilha deverá ser um valor inteiro x

- Seu programa deverá receber comandos que indicarão qual operação deve ser realizada (entre as quatro possibilidades) ou caso deseja terminar a execução do programa

- Lembre que ao fim do programa, você deverá liberar toda a memória alocada para a pilha

Trabalho

- Individual
- Entrega dia 27/10
- Peso 1