

# Estruturas de Dados I

## com Códigos C++

**Prof. Igor Machado Coelho**  
**igor.machado@ime.uerj.br**

**Departamento de Informática e Ciência da Computação**  
**Universidade do Estado do Rio de Janeiro**

**IME-04-10820 2015/2**



# Conteúdo do Curso

Pilhas

Pilha Sequencial

Pilha Encadeada



# Pilha

- ▶ Uma Pilha pode ser imaginada como vemos no cotidiano.
- ▶ Em uma pilha de pratos, por exemplo, só se consegue “inserir” (empilhar) novos pratos no topo da pilha, bem como “remover” (desempilhar) os pratos do topo.
- ▶ As chamadas de uma função recursiva podem ser simuladas utilizando uma pilha, e é na verdade desta maneira que o sistema operacional consegue executar tais funções!



# Pilha

- ▶ Uma Pilha é uma estrutura de dados linear (assim como as Listas) para armazenamento e busca de informações, consistindo de 3 operações básicas: topo, empilhar (push) e desempilhar (pop).
- ▶ Uma Pilha pode ser implementada utilizando uma lista linear, porém o acesso aos elementos de uma pilha estão restritos a apenas **uma** extremidade da lista.
- ▶ Seu comportamento é descrito como LIFO (last-in first-out), ou seja, o último elemento a entrar será o primeiro a sair.
- ▶ Assim como as listas, para o T.A.D. Pilha, estudaremos duas formas distintas de implementação: Sequencial e Encadeada.



# Pilhas Sequenciais

As Pilhas Sequenciais utilizam um array para armazenar as informações, então as informações sempre estarão em um espaço contíguo de memória.



# Pilha Sequencial – Definição

Consideraremos uma pilha com elementos do tipo caractere.

```
#define MAXN = 100000
class PilhaSeq1 {
    public:
        char elementos[MAXN];
        int N; // numero de elementos atualmente na pilha

        void constroi() {...}
        void destroi() {...}

        char topo() {...}
        void empilhar(char valor) {...}
        char desempilhar() {...}
}
```

Qual a complexidade de espaço da pilha sequencial?



# Pilha Sequencial – Utilização

Vamos testar a Pilha Sequencial para aprender na prática!

```
int main(){
    PilhaSeq1 P;
    P.constroi();
    P.empilhar('A');
    P.empilhar('B');
    P.empilhar('C');
    printf("%c\n", P.topo());
    printf("%c\n", P.desempilhar());
    P.empilhar('D');
    while(P.N > 0)
        printf("%c\n", P.desempilhar());
    P.destroi();

    return 0;
}
```

Quais são as impressões em tela?



# Pilha Sequencial – Construção e Destruição

A operação *constroi* inicializa a pilha para uso, e a função *destroi* libera os recursos requisitados.

```
class PilhaSeq1 {  
    ...  
  
    void constroi(){  
        N = 0;  
    }  
  
    void destroi(){  
        // vazio  
    }  
}
```

Qual a complexidade dos métodos acima?





# Pilha Sequencial – Topo

A operação de topo em uma pilha sequencial retorna o último elemento empilhado.

```
class PilhaSeq1 {  
    ...  
    void topo(){  
        return elementos[N-1];  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a pilha esteja vazia.



# Pilha Sequencial – Empilhar (Push)

A operação de empilhar em uma pilha sequencial adiciona um novo elemento ao topo da pilha.

```
class PilhaSeq1 {  
    ...  
    void empilhar(char valor){  
        elementos[N] = valor;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a pilha esteja cheia.



# Pilha Sequencial – Desempilhar (Pop)

A operação de desempilhar em uma pilha sequencial remove e retornar o último elemento da pilha.

```
class PilhaSeq1 {  
    ...  
    char desempilhar(){  
        N=N-1;  
        return elementos[N];  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a pilha esteja vazia.



# Pilha Sequencial – Limitações

A Pilha Sequencial tem a vantagem de ser bastante simples de implementar e ter complexidade  $O(1)$  e todas operações.

Porém, existe a limitação física de  $MAXN$  posições imposta pela alocação estática.

**Desafio:** implemente uma Pilha Sequencial utilizando **alocação dinâmica** para o vetor elementos. Assim, quando não houver espaço para empilhar novos elementos, crie mais espaço na memória (de preferência com o dobro do tamanho anterior) e mova o conteúdo atual da pilha para esse novo vetor. Dica: você vai precisar de alterar os métodos `constroi()`, `destrói()` e também inserir uma nova variável na Pilha para permitir a variação do limite máximo de elementos.

Iremos agora estudar uma implementação de pilha sem esta limitação de espaço máximo...



## Pilha Encadeada – Definição T.A.D.

```
class NoPilha1 {  
    char valor;  
    NoPilha1* prox;  
}  
  
class PilhaEnc1 {  
    public:  
    NoPilha1* inicio;  
    int N; // numero de elementos atualmente na pilha  
  
    void constroi(){...}  
    void destroi(){...}  
  
    char topo(){...}  
    void empilhar(char valor){...}  
    char desempilhar(){...}  
}
```

Qual a complexidade de espaço?



# Pilha Encadeada – Construção e Destruição

```
class PilhaEnc1 {  
    ...  
    void constroi(){  
        N = 0;  
    }  
  
    void destroi(){  
        while(inicio != NULL){  
            NoPilha1* p = inicio ->prox;  
            delete inicio;  
            inicio = p;  
        }  
    }  
}
```

Qual a complexidade dos métodos acima?



# Pilha Encadeada – Empilhar (Push)

```
class PilhaEnc1 {  
    ...  
    void empilhar(char v){  
        NoPilha1* no = new NoPilha1;  
        no->valor = v;  
        no->prox = inicio;  
        inicio = no;  
        N=N+1;  
    }  
}
```

Qual a complexidade do método acima?



## Pilha Encadeada – Desempilhar (Pop)

```
class PilhaEnc1 {  
    ...  
    char desempilhar(){  
        NoPilha1* p = inicio->prox;  
        char r = inicio->valor;  
        delete inicio;  
        inicio = p;  
        N=N-1;  
        return r;  
    }  
}
```

Qual a complexidade do método acima? **Tarefa:** imprima uma mensagem de erro caso a pilha esteja vazia.





# Pilha – Revisão

- ▶ Quais são os 3 métodos de uma pilha?
- ▶ Qual é a complexidade de cada método em uma Pilha Sequencial?
- ▶ Qual é a complexidade de cada método em uma Pilha Encadeada?
- ▶ Quais as vantagens e desvantagens de cada implementação de pilha?

**Desafio:** implemente uma pilha utilizando uma lista duplamente encadeada.



# Bibliografia

Szwarcfiter, J.L; Markenzon, L. Estruturas de Dados e seus Algoritmos. Rio de Janeiro, LTC, 1994.

## **Bibliografia Adicional:**

- ▶ Cerqueira, R.; Celes, W.; Rangel, J.L. Introdução a estruturas de dados: com técnicas de programação em C. Editora, 2004.
- ▶ Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein Algoritmos: Teoria e Prática. Ed. Campus, 2002.
- ▶ Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. Introduction to Algorithms, 3rd ed.. The MIT Press, 2009.
- ▶ Preiss, B.R. Estruturas de Dados e Algoritmos Ed. Campus, 2000;
- ▶ Knuth, D.E. The Art of Computer Programming - Vols I e III. 2nd Edition. Addison Wesley, 1973.
- ▶ Graham, R.L., Knuth, D.E., Patashnik, O. Matemática Concreta. Segunda Edição, Rio de Janeiro, LTC, 1995.