



Ciência da Computação

Campus Arapiraca



Aula 05: Pilhas

Prof. Dr. Rodolfo Carneiro
rodolfo.cavalcante@arapiraca.ufal.br



Introdução

Estruturas de dados básicas

- Conjuntos são fundamentais para a computação assim como o são
- para a matemática
- Conjuntos manipulados por algoritmos podem crescer, encolher e
- sofrer outras mudanças
- Algoritmos exigem vários tipos diferentes de operações sobre
- conjuntos
- Estruturas de dados descrevem conjuntos de dados relacionados entre si
- O estudo de estruturas de dados é parte fundamental para o desenvolvimento de algoritmos
- Programas são compostos por algoritmos e estruturas de dados



Introdução

- As operações sobre conjuntos dinâmicos podem ser agrupadas em duas categorias
 - consultas – retornam informações sobre os conjuntos
 - modificações – alteram o conjunto
- Operações típicas
 - BUSCA(x) – busca por um elemento com valor de chave x e retorna uma posição ou um ponteiro para o elemento
 - INSERE (x) – insere o elemento x no conjunto
 - REMOVE (x) – remove o elemento x do conjunto
 - MINIMO() – retorna o elemento do conjunto com a menor chave
 - MAXIMO() – retorna o elemento do conjunto com a maior chave
 - VAZIO() – verifica se o conjunto está vazio



Introdução

- A escolha de uma estrutura afeta:
 - quantidade memória para armazenamento
 - tempo de processamento
- Conhecer várias estruturas é importante
 - conhecer vantagens e aplicabilidade de cada estrutura
- Hoje estudaremos pilhas!



Introdução

- Algumas aplicações requerem inserções, retiradas e acessos em apenas um dos extremos da estrutura de dados
- Exemplos:
 - implementação de recursividade pelo SO
 - mecanismo de fazer/desfazer de editores de texto
 - Navegação de páginas Web
- Pilhas (stack) são estruturas de dados LIFO
 - last-in, first-out
 - ultimo elemento a entrar é o primeiro a sair



Introdução

- Analogia: pilha de pratos
 - inserção ou remoção de pratos na pilha é mais conveniente na parte superior
 - o prato mais recentemente inserido sempre fica no topo, acima dos demais
 - o primeiro a entrar é o último a ser lavado
- Existe uma ordem linear para pilhas, que é a ordem do “mais recente para o menos recente”
- Essa propriedade torna a pilha uma ferramenta ideal para processamento de estruturas aninhadas
 - subestruturas mais internas devem ser processadas antes da estrutura que as contenham



Introdução

- Principais operações do TAD pilha:
 - Criar uma pilha vazia
 - Verificar se a pilha está vazia
 - Empilhar o item x no topo da pilha
 - Desempilhar o item x que está no topo da pilha
 - Verificar o tamanho atual da pilha



Introdução

- Existem várias opções de estruturas de dados que podem ser usadas para implementar pilha
- As mais utilizadas são:
 - implementação por meio de arranjos (vetor)
 - implementação por meio de estruturas auto-referenciadas



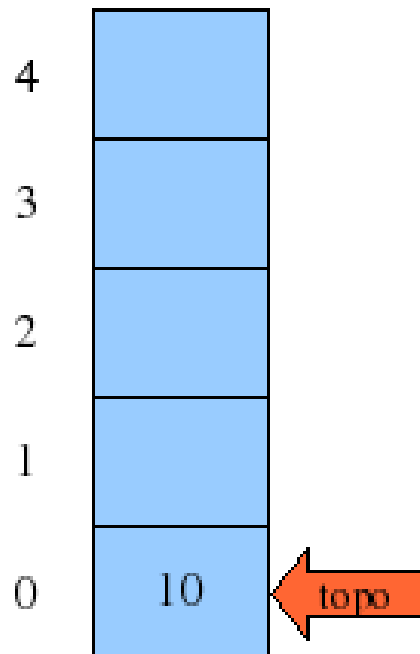
Pilhas com Arranjos

- Utilização de um vetor para armazenar itens
- Itens são armazenados em posições consecutivas de memória
- Inserções e retiradas acontecem na última posição do vetor (topo da pilha)
- Vetor deve ter tamanho suficiente para armazenar itens da pilha



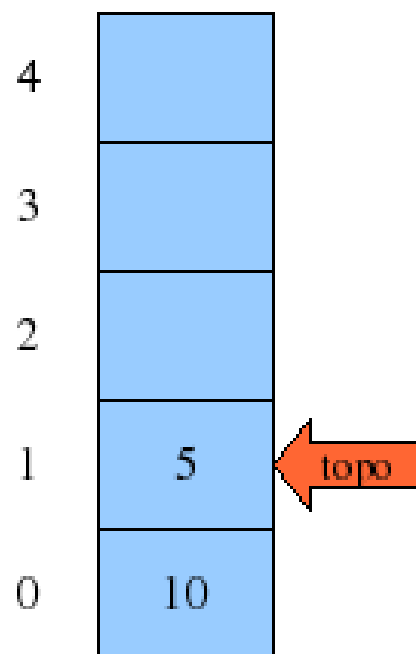
Pilhas com Arranjos

Empilhar (10)



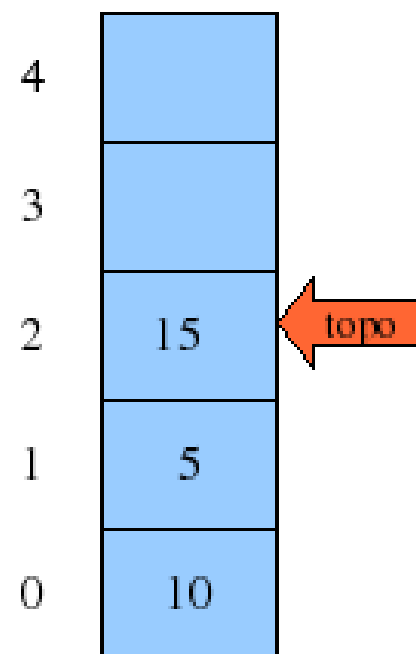
Topo: 0

Empilhar (5)



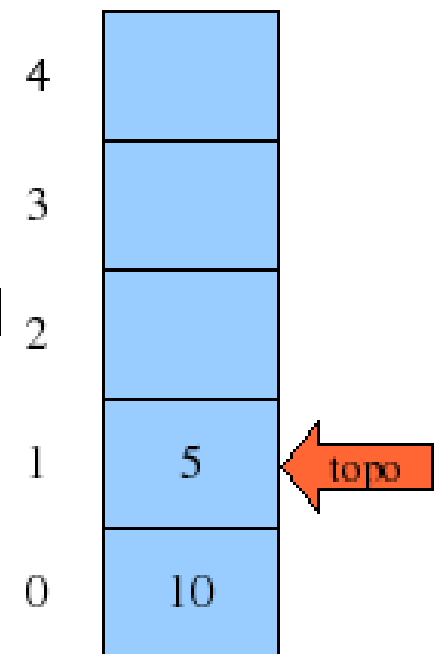
Topo: 1

Empilhar (15)



Topo: 2

Desempilhar



Topo: 1



Pilhas com Arranjos

- Implementação da pilha
 - Construtor
 - Verificar se está vazia
 - Empilhar
 - Desempilhar
 - Imprimir
 - Ver tamanho



Pilhas com Arranjos

Aplicação

- Verificar se uma expressão aritmética com parênteses e colchetes é bem formada
- Parênteses e colchetes são fechados na ordem inversa que são abertos
- Ex: expressão bem formada
 - $3-[15+2*(4-3)*[2+(5-1)]]/4$
- Ex: expressão mal formada
 - $5-[4+(0-3]$
- Como resolver o problema usando pilhas?



Pilhas com Arranjos

Aplicação

- Como resolver o problema usando pilhas?
 - Sempre que encontrar um parêntese ou colchete abrindo, empilhe-o
 - Quando encontrar parêntese ou colchete fechando, desempilhe o símbolo do topo da pilha
 - Compare se eles são equivalentes



Pilhas com Arranjos

Considerações sobre implementação de pilhas com arranjos

- Economia de memória – apontadores são implícitos e não precisam ser armazenados na memória
- Problema: tamanho da pilha é fixo
 - aplicações que não possuem previsão sobre crescimento da pilha requerem realocação de memória
 - operação de alto custo de tempo e memória
 - criar nova área com mais posições e copiar todos os itens para esta área
- O que fazer quando o crescimento da pilha é dinâmico e não previsto?



Estruturas Auto-Referenciadas

- Pilha é organizada em células
- Cada célula guarda um item de dado e uma referência para a próxima célula
- Permite utilizar posições não contíguas de memória
- Não é necessário definir a priori o tamanho máximo da pilha
- A pilha guarda uma referência para a célula no topo
- Cada células aponta para a próxima célula



Estruturas Auto-Referenciadas

- Implementação



Estruturas Auto-Referenciadas

- Considerações sobre estruturas auto-referenciadas
 - Implementação útil em aplicações em que não existe previsão sobre o crescimento da pilha
- Desvantagem
 - utilização de memória extra para armazenar as referências (os ponteiros prox)



Exercícios

1. Uma palavra ou frase é um palíndromo quando pode ser lida de trás pra frente mantendo o mesmo sentido. Exemplos de palavras são “ovo”, “radar”, “reviver”. Exemplos de frases são “A base do teto desaba”, “A dama admirou o rim da amada”.

Faça um programa que recebe uma palavra ou frase e responda se ela é um palíndromo ou não.



Exercícios

2. Um problema enfrentado por uma empresa que gerencia um porto é o empilhamento de containers. Uma pilha de containers pode ter até 5 containers empilhados. Quando um novo container precisa ser empilhado, deve-se empilhá-lo na menor pilha no momento. Cada container possui um código associado. No porto é possível fazer apenas 4 pilhas de containers, mais uma pilha que pode ser utilizada temporariamente (swap). Essa pilha adicional deve ser mantida vazia e utilizada apenas caso um container que não está no topo da pilha precise ser retirado. Faça um algoritmo que recebe uma lista de containers e imprima o resultado das pilhas ao final da operação.