

Estruturas de Dados

Pilhas, Filas e Deques

Prof. Marcos Caetano

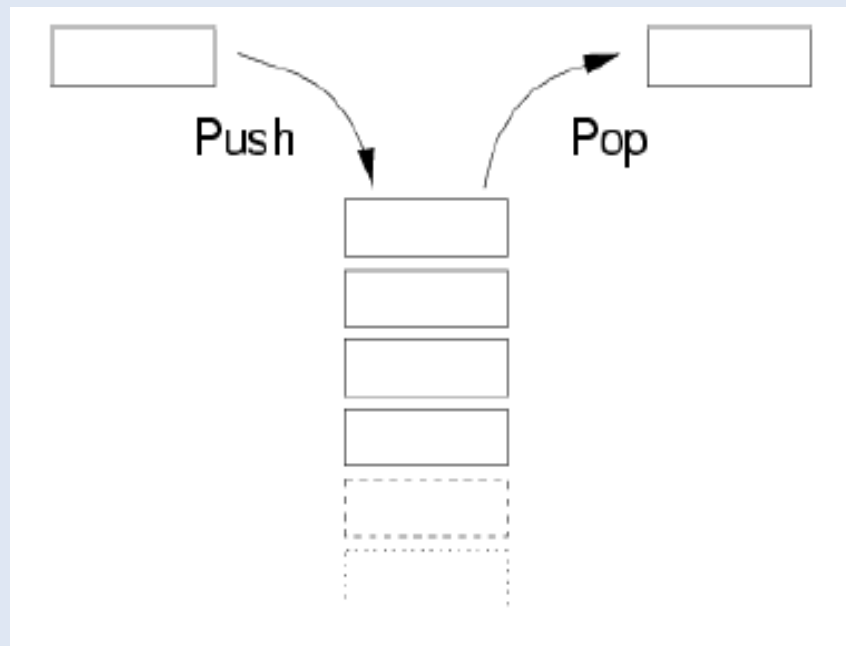
(Material Base – Prof. Eduardo Alchieri)

Estruturas de Dados

Pilhas

Pilhas

- Lista LIFO (Last In, First Out)
- Os elementos são colocados na estrutura (pilha) e retirados em ordem inversa a de sua chegada
 - Inserção apenas no topo da pilha
 - Remoção apenas no topo da pilha



Pilhas

- Operações possíveis:
 - Esvaziar a pilha
 - Verificar se a pilha está vazia
 - Colocar um dado no topo da pilha (empilhar)
 - Retirar o dado do topo da pilha (desempilhar)
 - Etc.

Pilhas

- Implementação de pilhas
 - Usando vetores
 - Pode-se implementar uma pilha de tamanho fixo usando vetores. Este tamanho determinará o número máximo de elementos que poderão estar na pilha ao mesmo tempo
 - É necessário um inteiro para armazenar o valor da posição do vetor aonde encontra-se o topo da pilha
 - Usando uma lista ligada
 - A implementação de uma pilha que usa como estrutura básica uma lista ligada é mais simples, pois a lista não é uma estrutura de tamanho fixo
 - Basicamente, os dados devem ser colocados (empilhados) em alguma das extremidades da lista e retirados (desempilhados) a partir desta mesma extremidade

Pilhas

- É ideal para processamento de estruturas aninhadas de profundidade imprevisível
- Uma pilha contém uma sequência de obrigações adiadas. A ordem de remoção garante que os dados mais internos serão processados depois dos mais externos
 - Quando é necessário percorrer um conjunto de dados e guardar uma lista de coisas a fazer posteriormente
 - O controle de sequências de chamadas de subprogramas
 - A sintaxe de expressões aritméticas

Pilhas

- Aplicações
 - Controle de delimitadores em uma equação (rastrear escopo): $\{(A + B) * C + (D + E) / [F + G] + H\} / I$
 - Algoritmo: Percorrer a equação da seguinte forma:
 - 1 – se um inicializador de escopo for encontrado, o mesmo é empilhado
 - 2 – se um finalizar de escopo for encontrado, a pilha é verificada
 - Se estiver vazia, então a equação é incorreta
 - Se não, desempilhar e comparar com o finalizadorAo final, a pilha deve estar vazia

Pilhas

- Aplicações
 - Recursividade
 - A solução de um problema depende da solução de instâncias menores do mesmo problema

Fatorial iterativo

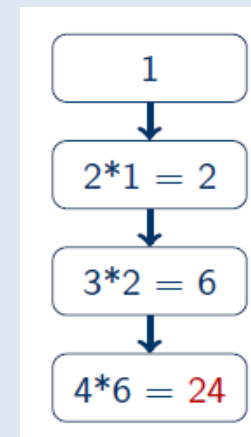
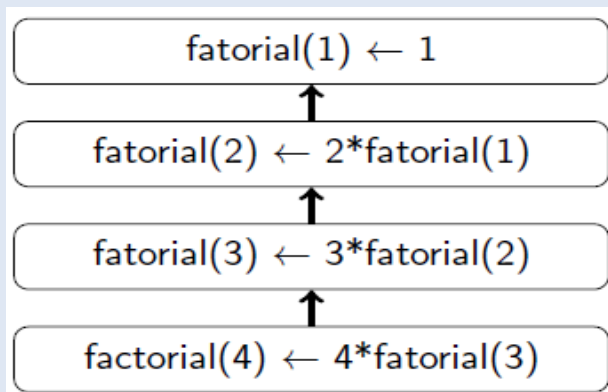
```
int factorial (int n) {  
    int result = 1;  
    while(n > 1) {  
        result *= n;  
        n -= 1;  
    }  
    return result ;  
}
```

Fatorial recursivo

```
int factorial (int n) {  
    if (n ≤ 1)  
        return 1;  
    else  
        return n * factorial (n-1);  
}
```


Pilhas

- Aplicações
 - Quanto é $\text{fatorial}(4)$?



Pilhas

- Aplicações
 - Avaliação de expressões posfixas (Trabalho 1)
 - Notação infixa (é a usual): $(5 + 9) * 2 + 6 * 5$
 - Os operadores binários aparecem entre os operandos
 - A ordem das operações são determinadas pela precedência dos operadores ou pelos parênteses "(" ")"
 - Notação prefixa (notação polonesa): $+ * + 5 9 2 * 6 5$
 - Notação posfixa (notação polonesa reversa): $5 9 + 2 * 6 5 * +$
 - Não faz uso de parênteses e não requer relações de precedência

Pilhas

- Aplicações
 - Algoritmo para avaliação de expressões posfixas
 - Os componentes da expressão são processados da esquerda para a direita como a seguir:
 - Se o próximo componente da expressão é um operando, o valor do componente é colocado na pilha
 - Se o próximo componente da expressão é um operador, então os seus operandos estão na pilha. O número requerido de operandos é retirado da pilha, a operação específica é realizada, e o resultado é armazenado de volta na pilha.
 - Ao final, a pilha conterá um único dado que é o valor final da expressão
 - Simule a execução do algoritmo com a expressão a seguir:

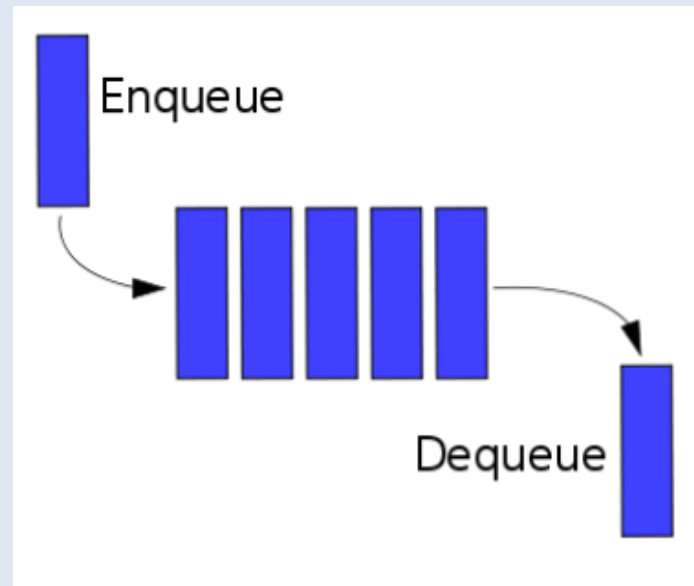
5 9 + 2 * 6 5 * +

Estruturas de Dados

Filas

Filas

- Lista FIFO (First In, First Out)
- Os elementos são colocados e retirados por ordem de chegada
 - Inserção apenas no final da fila
 - Remoção apenas no início da fila



Filas

- Uma fila permite várias operações
 - Criar uma fila vazia
 - Inserir um novo item (no final)
 - Remover um item (do início)
 - Esvaziar a fila
 - Etc.

Filas

- Implementação de filas
 - Usando vetores
 - Pode-se implementar uma fila de tamanho fixo usando vetores. Este tamanho determinará o número máximo de elementos que poderão estar na fila ao mesmo tempo
 - É necessário dois inteiros para armazenar o valor das posições do vetor aonde se encontram o início e o final da fila
 - Usando uma lista ligada
 - A implementação de uma fila que usa como estrutura básica uma lista ligada é mais simples, pois a lista não é uma estrutura de tamanho fixo
 - Basicamente, os dados devem ser colocados (enfileirados) no final da lista e retirados (desenfileirados) do início da lista.

Filas

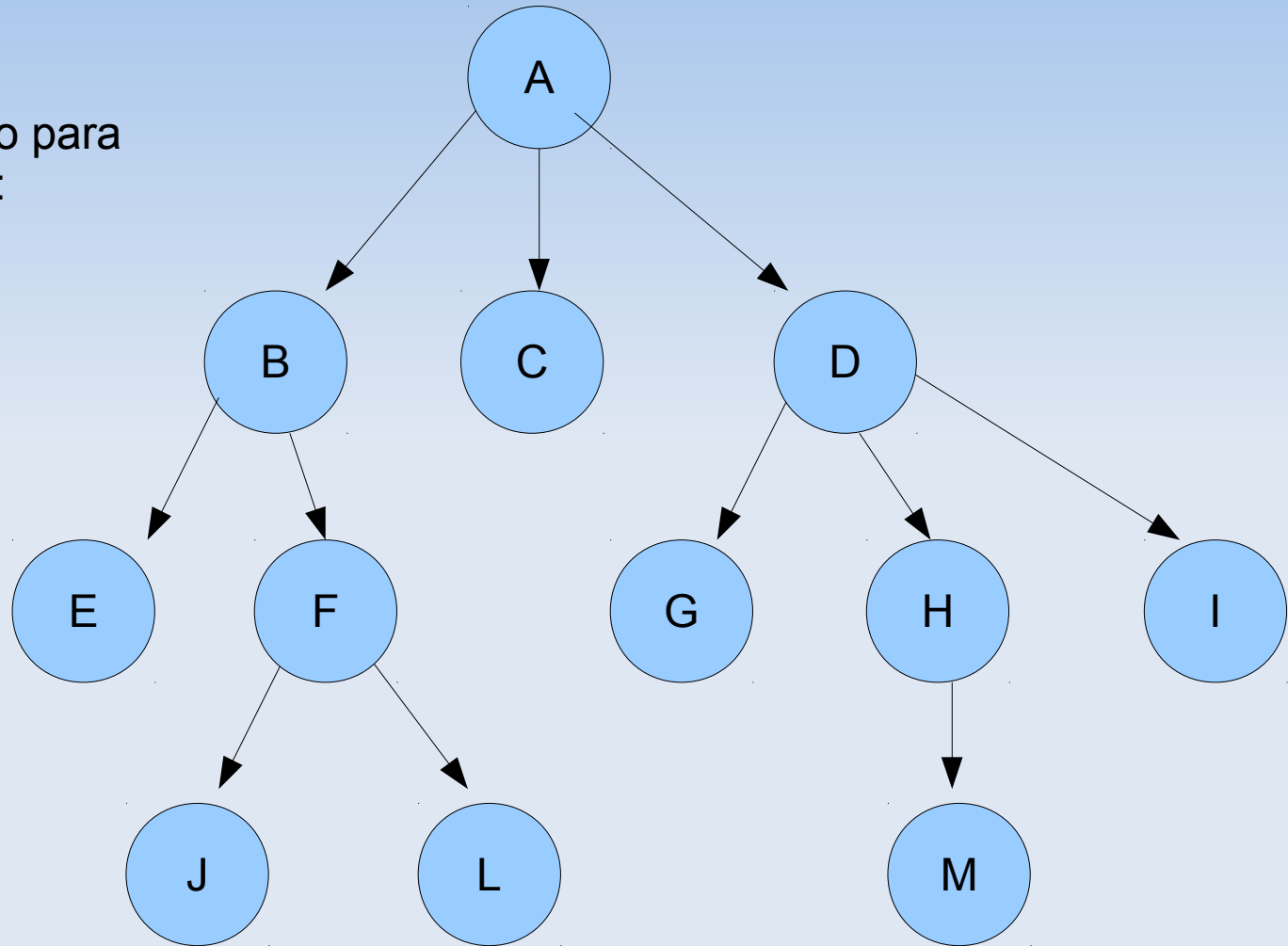
- Aplicações
 - As filas são utilizadas quando desejamos processar itens de acordo com a ordem de chegada (o primeiro a chegar será o primeiro a ser processado).
 - Sistemas operacionais, por exemplo, usam filas para regular a ordem em que as tarefas devem receber processamento e recursos devem ser alocados a processos.
 - Fila de prioridade (inserção ordenada)
 - Outras aplicações
 - Manipulação de uma sequência de caracteres

Filas

- Aplicações
 - Utilizada na implementação de percurso em largura em árvores
 - Um percurso em largura percorre (visita) os nós da árvore segundo a ordem de seus níveis
 - Uma maneira de implementar um percurso em largura de uma árvore é através do uso de uma fila:
 - Para começar o percurso, o nó raiz é posto na fila
 - Após, repetir os seguintes passos até que a fila esteja vazia:
 - Retire e visite (percorra) o primeiro nó da fila
 - Coloque, da esquerda para a direita, seus filhos na fila

Filas

Simule o algoritmo para
a seguinte árvore:



Pilhas e Filas

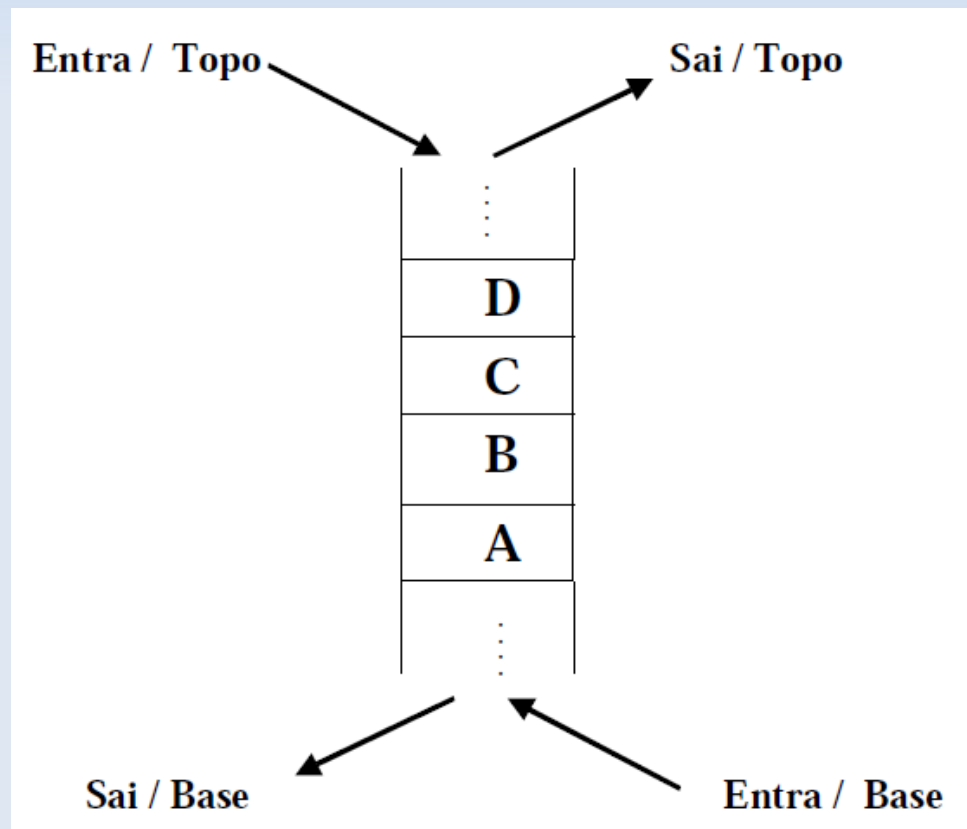
- Exercício
 - Dada uma lista de caracteres formada por uma seqüência alternada de letras e dígitos, construa um método que retorne uma lista na qual as letras são mantidas na seqüência original e os dígitos são colocados na ordem inversa
 - Exemplo:
 - A 1 E 5 T 7 W 8 G → A 8 E 7 T 5 W 1 G
 - 3 C 9 H 4 Q 6 → 6 C 4 H 9 Q 3
 - Suponha a existência de uma função `ehDigito(ch caractere)` que retorna 1 caso o *caractere* seja um dígito e 0 caso contrário.

Estruturas de Dados

Dequeues

Dequeues

- *Double-ended queue*
- Extensão de filas que permite inserir e remover dados em ambas as extremidades



Dequeues

- Uma deque permite várias operações
 - Criar uma deque
 - Inserir um novo item no início
 - Inserir um novo item no final
 - Remover o item do início
 - Remover o item do final
 - Esvaziar
 - Etc.

Dequeues

- A implementação de dequeues também poder ser por meio de vetores ou listas ligadas
- Na implementação através de listas ligadas, a utilização de uma lista duplamente encadeada torna a implementação mais eficiente
 - Remover o nó do final é mais eficiente em uma lista duplamente encadeada do que em uma lista simplesmente encadeada