

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

Estrutura de Dados

Listas

Por Juliano P. Menzen

Sumário

- ▶ Introdução;
- ▶ Fundamentos;
- ▶ Tipos de Lista;
- ▶ Alocação de Memória;
- ▶ Listas Encadeadas;
- ▶ Dúvidas;
- ▶ Exercícios;

Introdução

- ▶ Permitem manter dados agrupados:
 - ▶ Lista de compras;
 - ▶ Lista de chamada;
- ▶ Em computação, a lista é uma das estruturas de dados mais empregadas no desenvolvimento de programas;
- ▶ Listas permitem resolver problemas em que conjuntos de dados se relacionam entre si de forma que este relacionamento reflete algo do mundo real e que precisa ser mantido para ser possível refletir esta necessidade do mundo real no mundo computacional. Exemplo:
 - ▶ Supondo que desejamos encontrar a precipitação pluviométrica média do período, visualizando cronologicamente o comportamento dos indicadores diários de chuva. Neste caso, é preciso que mantenhamos um relacionamento entre elementos de dados anteriores e posteriores de tal modo a obter uma visão do todo e do comportamento médio.

Fundamentos

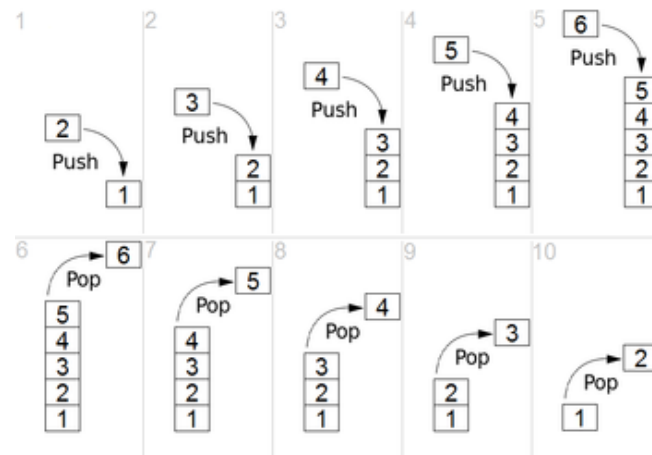
- ▶ Uma **Lista Linear** é uma estrutura que permite representar um conjunto de dados de forma a preservar a relação de ordem linear (ou total) entre eles;
- ▶ Uma lista é composta de **nós**, os quais podem conter, cada um deles, um registro composto de um ou mais tipos de dados;
- ▶ Ou seja, uma lista é formada por um conjunto ou coleção de $n \geq 0$ de **nós** tal que:
 - ▶ $L: [x_1, x_2, \dots, x_n]$, onde x_k é precedido por x_{k-1} e seguido de x_{k+1} , para $1 < k < n$.
 - ▶ Se $n=0$ diz-se que a lista é vazia.
 - ▶ x_1 é o primeiro elemento de L ;
 - ▶ x_n é o último elemento de L ;
 - ▶ x_k , $1 < k < n$, é precedido pelo elemento x_{k-1} e seguido por x_{k+1} em L .
- ▶ A característica fundamental de uma lista linear é o sentido de ordem unidimensional dos elementos que a compõem. Uma ordem que nos permite dizer com precisão onde a coleção inicia-se e onde termina, sem possibilidade de dúvida.

Fundamentos

- ▶ Uma Lista permite diversas operações a serem conhecidas abaixo:
 - ▶ *acessar* um elemento qualquer da lista;
 - ▶ *inserir* um elemento numa posição específica da lista;
 - ▶ *remover* um elemento de uma posição específica da lista;
 - ▶ *combinar* duas lista em uma única;
 - ▶ *particionar* uma lista em duas;
 - ▶ obter *cópia* de uma lista;
 - ▶ determinar o *total* de elementos na lista;
 - ▶ *ordenar* os elementos da lista;
 - ▶ *procurar* um determinado elemento na lista;
 - ▶ *apagar* uma lista;
 - ▶ outras...

Tipos de lista

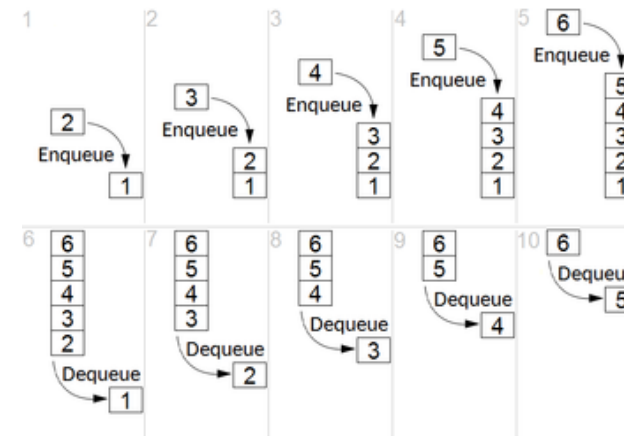
- ▶ **PILHA:**
- ▶ Lista linear onde todas as inserções, remoções e acessos são realizados em um único extremos.
- ▶ Lista com esta característica são também denominadas listas **LIFO** (Last-In/Frist-Out, ou em português: último que entra/primeiro que sai).
- ▶ Dentre os elementos que permanecem no conjunto, o primeiro elemento a ser retirado é o último que foi inserido;



Fonte: wikipedia

Tipos de lista

- ▶ **FILA:**
- ▶ Lista linear onde todas as inserções são feitas num certo extremo e todas as remoções e acessos são realizadas no outro.
- ▶ Filas são também denominadas listas **FIFO** (Frist-In/Frist-Out, ou em português: primeiro que entra/primeiro que sai).
- ▶ Dentre os elementos que ainda permanecem no conjunto, o primeiro elemento a ser retirado é o primeiro que foi inserido.

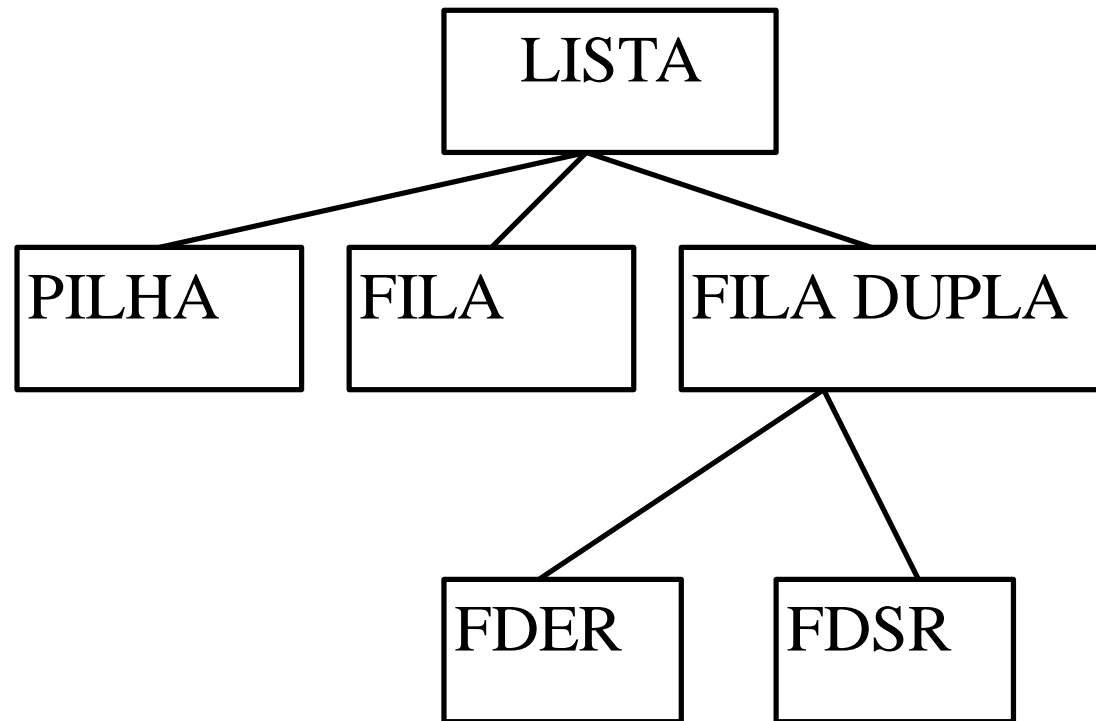


Fonte: wikipedia

Tipos de lista

- ▶ **FILA DUPLA:**
- ▶ Lista Linear onde as inserções, remoções ou acessos são realizados em qualquer extremo.
- ▶ Filas Duplas são também denominadas **DEQUE** (Double-Ended QUEue, ou em português: fila de extremidade dupla).
- ▶ Uma Fila Dupla pode ainda gerar dois casos especiais:
 - ▶ Fila Dupla de Entrada Restrita (se a inserção for restrita a um único extremo);
 - ▶ Fila Dupla de Saída Restrita (se a remoção for restrita a um único extremo);

Tipos de lista



Alocação de memória

- Existem quatro categorias para alocação de memória, sendo que cabe ao programador escolher quais usar em seus programas:

	Seqüencial	Encadeada
Estática	Estática Seqüencial	Estática Encadeada
Dinâmica	Dinâmica Seqüencial	Dinâmica Encadeada

Alocação de memória

- ▶ **Alocação Estática:**

- ▶ Quando a quantidade total de memória utilizada pelos dados é previamente conhecida e definida no código fonte do programa e durante toda a sua execução, a quantidade de memória utilizada pelo programa não varia. Ex:

- ▶ `Int vetor[5];`

- ▶ **Alocação Dinâmica:**

- ▶ Quando o programa é capaz de criar novas variáveis enquanto executa, isto é, se áreas de memória que não foram declaradas no programa passam a existir durante a sua execução. Ex:

- ▶ `Int *vetor;`

- ▶ `Vetor = (*int)malloc(sizeof(int)*5);`

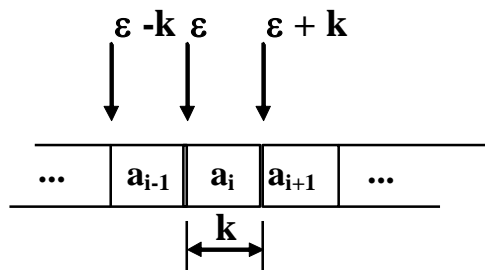
- ▶ Ou

- ▶ `Vetor = (*int)calloc(5, sizeof(int));`

Alocação de memória

► Alocação Sequencial:

- Armazena os elementos de uma lista em células consecutivas de memória, um após o outro.



- Endereço(a_i) = ε
- Endereço(a_{i-1}) = $\varepsilon - k$
- Endereço(a_{i+1}) = $\varepsilon + k$
- A maior vantagem no uso de uma área sequencial de memória para armazenar uma lista linear é que, dado o endereço inicial β da área alocada e o índice i de um elemento qualquer da lista, podemos acessá-lo imediatamente, com um simples e rápido cálculo;
- O ponto fraco desta forma de armazenamento aparece quando precisamos inserir ou suprimir elementos do meio da lista, quando então um certo esforço será necessário para movimentar os elementos, de modo a abrir espaço para inserção, ou de modo a ocupar o espaço liberado por um elemento que foi removido.

Alocação de memória

► Alocação Encadeada:

- Ao invés de manter os elementos agrupados numa área contínua de memória, na alocação encadeada os elementos podem ocupar quaisquer células (não necessariamente consecutivas) e, para manter a relação da ordem linear, juntamente com cada elemento é armazenado o endereço do próximo elemento da lista;
- Na alocação encadeada, os elementos são armazenados em blocos de memória denominados **nodos**, sendo que cada **nodo** é composto por dois campos:
 - um para armazenar dados;
 - outro para armazenar endereço.
- Dois endereços especiais devem ser destacados:
 - o endereço do primeiro elemento da lista (L);
 - o endereço do elemento fictício que segue o último elemento da lista (ϕ).

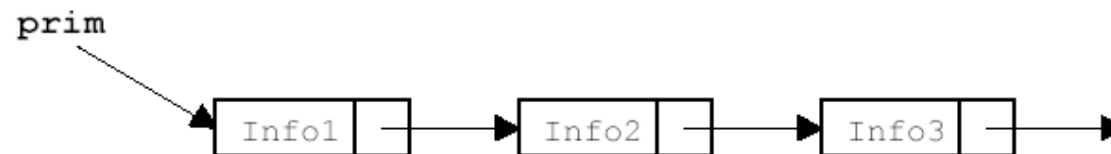
Alocação de memória

► Alocação Encadeada:

- A alocação encadeada apresenta como maior vantagem a facilidade de inserir ou remover elementos do meio da lista. Como os elementos não precisam estar armazenados em posições consecutivas de memória, nenhum dado precisa ser movimentado, bastando atualizar o campo de ligação do elemento que precede aquele inserido ou removido;
- A grande desvantagem da alocação encadeada surge quando desejamos acessar uma posição específica dentro da lista. Neste caso devemos partir do primeiro elemento e ir seguindo os campos de ligação, um a um, até atingir a posição desejada. Para listas extensas, esta operação pode ter um alto custo em relação a tempo;

Lista Encadeada

- ▶ Numa lista encadeada, para cada novo elemento inserido na estrutura, alocamos um espaço de memória para armazená-lo. Desta forma, o espaço total de memória gasto pela estrutura é proporcional ao número de elementos nela armazenado;
- ▶ A estrutura consiste numa sequência encadeada de elementos, em geral chamados de nós da lista. A lista é representada por um ponteiro para o primeiro elemento (ou nó);
- ▶ Do primeiro elemento, podemos alcançar o segundo seguindo o encadeamento, e assim por diante. O último elemento da lista aponta para NULL, sinalizando que não existe um próximo elemento;



Lista Encadeada

- ▶ Exemplo:
- ▶ Ver arquivo “lista encadeada.cpp”

Dúvidas



Exercícios

- ▶ 1) Por que motivo as listas devem ser implementadas na forma seqüencial quando acessar um elemento arbitrário for uma operação muito requisitada?
- ▶ 2) Usando C, como você faria uma alocação estática seqüencial de uma área de memória suficiente para armazenar 10 elementos do tipo real?
- ▶ 3) Usando C, como você definiria uma variável capaz de representar um nodo de uma lista linear implementada com alocação encadeada?
- ▶ 4) Por que motivo os valores L e ϕ são considerados especiais no esquema de alocação encadeada apresentada no material?
- ▶ 5) Criar 2 listas com valores int, não ordenados, simplesmente encadeados, solicitando o numero de elementos de cada lista. Apresente um algoritmo que intercale as duas listas, de forma que a lista resultante contenha todos os elementos das duas listas.