

Listas Lineares

Prof. Flavio B. Gonzaga
flavio.gonzaga@unifal-mg.edu.br
Universidade Federal de Alfenas
UNIFAL-MG

Sumário

- Listas lineares
- Alocação sequencial
 - Operações:
 - Busca;
 - Busca em lista ordenada;
 - Busca binária;

Listas lineares

- Listas lineares são as estruturas de dados de manipulação mais simples;
- Uma lista linear agrupa informações referentes a um conjunto de elementos que, de alguma forma, relacionam entre si.

Listas lineares

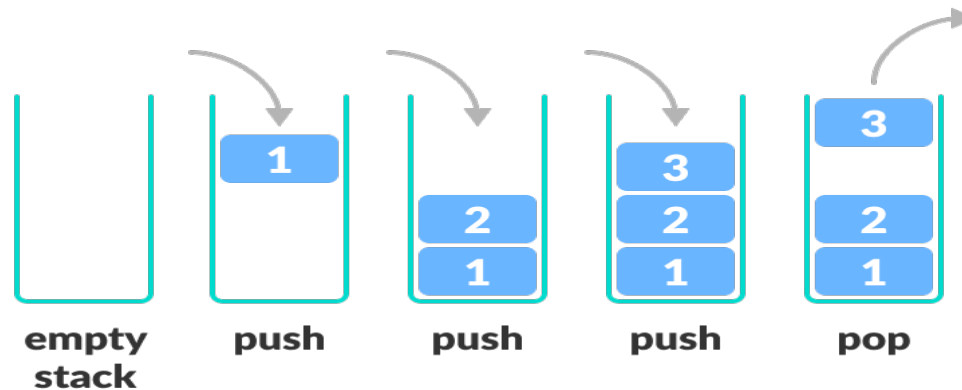
- Uma *lista linear*, é então um conjunto de $n \geq 0$ nós, $L[1]$, $L[2]$, ..., $L[n]$ tais que suas propriedades estruturais decorrem, unicamente, da posição relativa dos nós dentro da sequência linear. Tem se:
 - Se $n > 0$, $L[1]$ é o primeiro nó,
 - Para $1 < k \leq n$, o nó $L[k]$ é precedido por $L[k-1]$.

Listas lineares

- As operações mais frequentes em listas são a busca, inclusão e remoção de um determinado elemento;
- Tais operações podem ser consideradas básicas, e por isso, é necessário que os algoritmos que as implementem sejam eficientes;

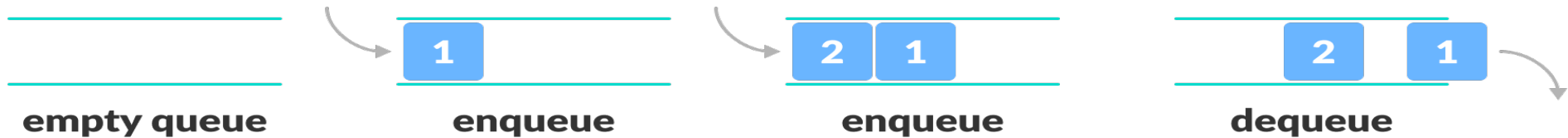
Listas lineares

- Existem alguns casos particulares de listas:
 - Inserções e remoções são realizadas somente em um extremo:
 - **Pilha.**



Listas lineares

- Existem alguns casos particulares de listas:
 - Inserções são realizadas em um extremo, e remoções no outro:
 - Fila.



Listas lineares

- Existem alguns casos particulares de listas:
 - Inserções e remoções são permitidas nas extremidades da lista:
 - *Deque (double ended queue)*.



Listas lineares

- O tipo de armazenamento de uma lista linear pode ser classificado de acordo com a posição relativa (sempre contígua ou não) na memória;
- O primeiro caso corresponde à *alocação sequencial* na memória, enquanto que o segundo é conhecido como *alocação encadeada*;

Alocação sequencial

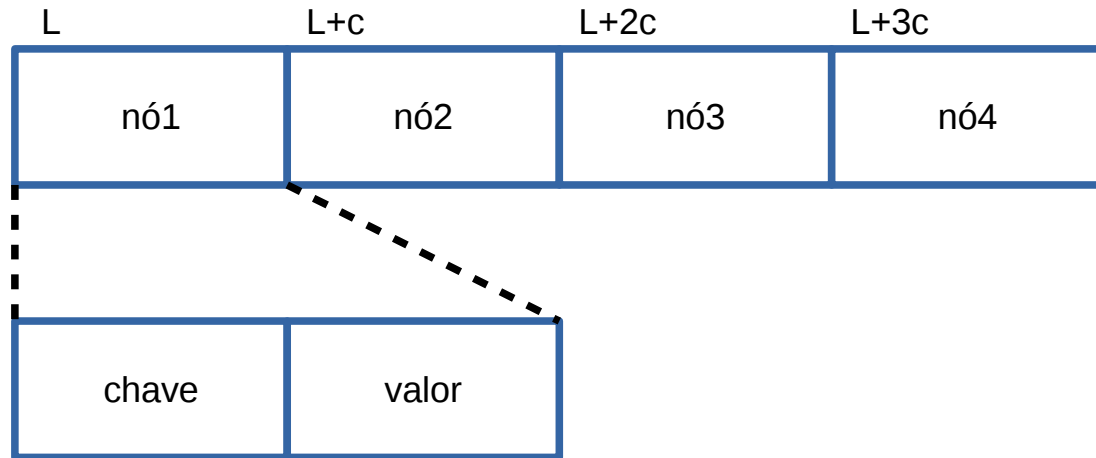
- Geralmente realizada com a reserva prévia de memória para cada estrutura utilizada;
 - A inserção e remoção não ocorrem de fato!
 - Utiliza-se de algum tipo de simulação para essas operações;
 - Considera-se portanto como uma *alocação estática*;

Alocação sequencial

- Seja uma lista linear, cada nó é formado por campos, que armazenam as características distintas dos elementos da lista;
- Cada nó possui, geralmente, um identificador, denominado *chave*;
- Para evitar ambiguidade, assume-se que todas as chaves são distintas;
- Os nós podem se encontrar ordenados, ou não, segundo os valores de suas chaves;
 - Assim a lista pode ser *ordenada* ou *não ordenada*;

Alocação sequencial

- Exemplo de um nó:



Alocação sequencial

- Busca de um elemento na lista L:

função *busca1(x)*

$i := 1; \quad busca1 := 0$

enquanto $i \leq n$ faça

se $L[i].chave = x$ então

$busca1 := i$

$i := n + 1$

senão $i := i + 1$

Alocação sequencial

- Busca de um elemento na lista L:

função *busca1*(*x*)

$i := 1; \quad busca1 := 0$

enquanto $i \leq n$ faça

se $L[i].chave = x$ então

$busca1 := i$

$i := n + 1$

senão $i := i + 1$

função *busca*(*x*)

$L[n+1].chave := x; \quad i := 1;$

enquanto $L[i].chave \neq x$ faça

$i := i + 1$

se $i \neq n + 1$ então

$busca := i$

senão $busca := 0$

Alocação sequencial

- Busca de um elemento na lista L, **ordenada**:

função *busca-ord*(x)

$L[n+1].chave := x; i := 1;$

enquanto $L[i].chave < x$ faça

$i := i + 1$

se $i = n + 1$ ou $L[i].chave \neq x$ então

$busca-ord := 0$

senão $busca-ord := i$

Alocação sequencial

- Busca de um elemento na lista L, **ordenada**:

função *busca-bin*(x)

$inf := 1; sup := n; busca-bin := 0;$

enquanto $inf \leq sup$ faça

$meio := \lfloor (inf + sup) / 2 \rfloor$

se $L[meio].chave = x$ então

$busca-bin := meio$

$inf := sup + 1$

senão se $L[meio].chave < x$

$inf := meio + 1$

senão $sup := meio - 1$

Referências Bibliográficas

- Estruturas de Dados e Seus Algoritmos. Szwarcfiter J. L.; Markenzon L.. 3a Edição. Editora LTC. 2010.
- Figura pilha: <https://www.programiz.com/dsa/stack> , acesso em 30/11/2020.
- Figura fila: <https://www.programiz.com/dsa/queue>, acesso em 30/11/2020.
- Figura deque: <https://www.programiz.com/dsa/deque>, acesso em 30/11/2020.