

Listas Lineares

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

Sumário

- Listas lineares
- Alocação encadeada (dinâmica)
 - Motivação
 - Características
- Lista simplesmente encadeada
 - Operações
 - Busca
 - Inserção
 - Remoção

Alocação encadeada

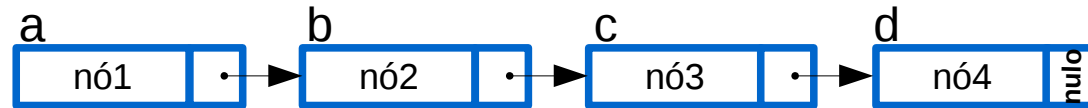
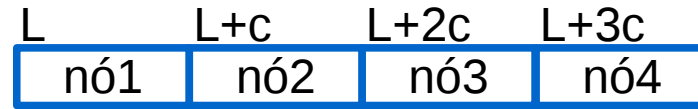
- Motivação:
 - De forma geral, o desempenho de algoritmos que implementam operações realizadas em listas com alocação sequencial tende a ser bem ruim.

Alocação encadeada

- Características:
 - A alocação encadeada (também conhecida como alocação dinâmica) se caracteriza pela alocação (ou liberação) de posições de memória conforme demanda da aplicação.
 - Há vantagens e desvantagens associadas a cada tipo de alocação.
 - Entretanto, só podem ser precisamente medidas ao se conhecerem as operações envolvidas na aplicação desejada.

Alocação encadeada

- Características:
 - Lista linear sequencial x encadeada.



Alocação encadeada

- Características:
 - De maneira geral pode-se afirmar que a alocação encadeada, apesar de ter um gasto de memória maior em virtude da necessidade de um novo campo no nó (o ponteiro para o próximo), é mais conveniente quando o problema inclui o tratamento de mais de uma lista.
 - Tal fato se aplica tanto à gerência do armazenamento quanto às operações em si, como:
 - Juntar listas.
 - Separar listas em sublistas.
 - Etc.
 - Por outro lado, o acesso ao k-ésimo elemento da lista é imediato em alocação sequencial, enquanto que na encadeada obriga ao percurso até o elemento desejado.

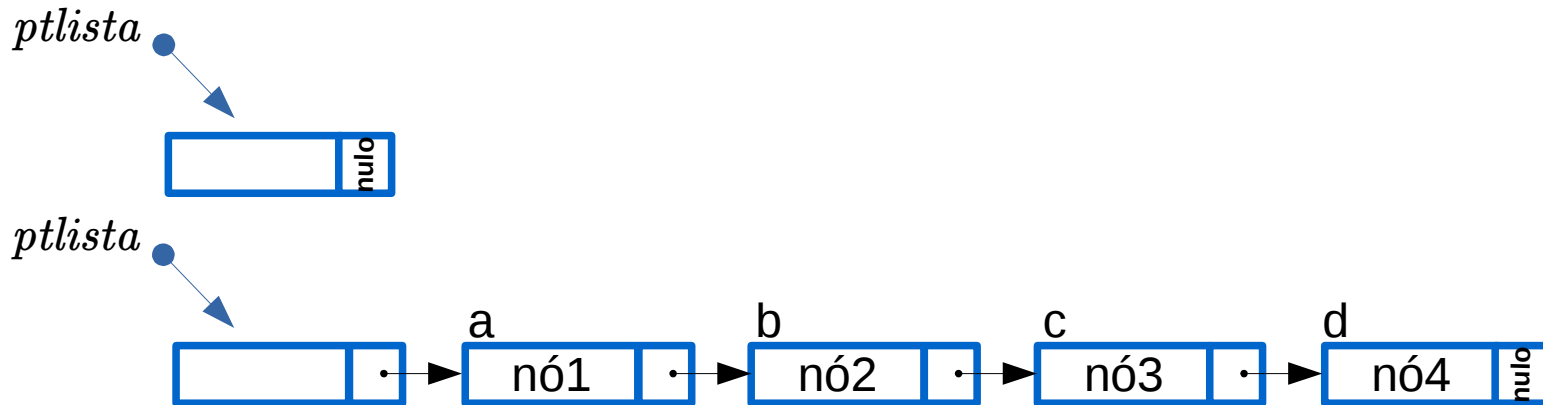
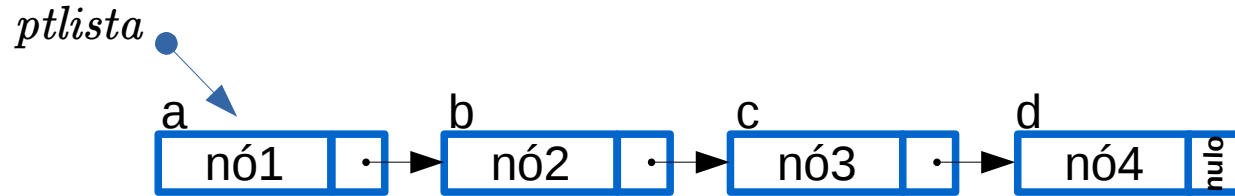
Alocação encadeada

- Características:
 - Qualquer estrutura que seja armazenada em alocação encadeada requer o uso de um ponteiro que indique o endereço de seu primeiro nó.
 - O percurso da lista é feito então a partir desse ponteiro.

Quando fizermos uso de ponteiro, a seguinte notação será usada: \uparrow
Assim, $pt\uparrow.info$ representa o campo *info* do nó apontado por *pt*.

Alocação encadeada

- Características:



Lista simplesmente encadeada

- Operações – busca.
 - Assim como na alocação sequencial, na alocação encadeada o **algoritmo de busca** também é utilizado nas operações de inserção, remoção, dentre outras.

Lista simplesmente encadeada

- Operações – busca.
- Funcionamento:
 - Assumindo uma lista **ordenada**;
 - *pont* terá o endereço do nó caso o mesmo seja encontrado;
 - *nulo* caso contrário;

procedimento *busca-enc*(*x*, *ant*, *pont*)

ant := *ptlista*; *pont* := *nulo*

ptr := *ptlista*↑.*prox*

enquanto *ptr* ≠ *nulo* faça

se *ptr*↑.*chave* < *x* então

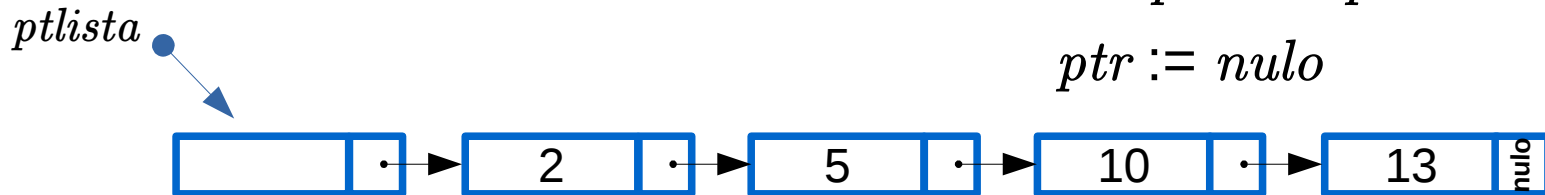
ant := *ptr*

ptr := *ptr*↑.*prox*

senão se *ptr*↑.*chave* = *x* então

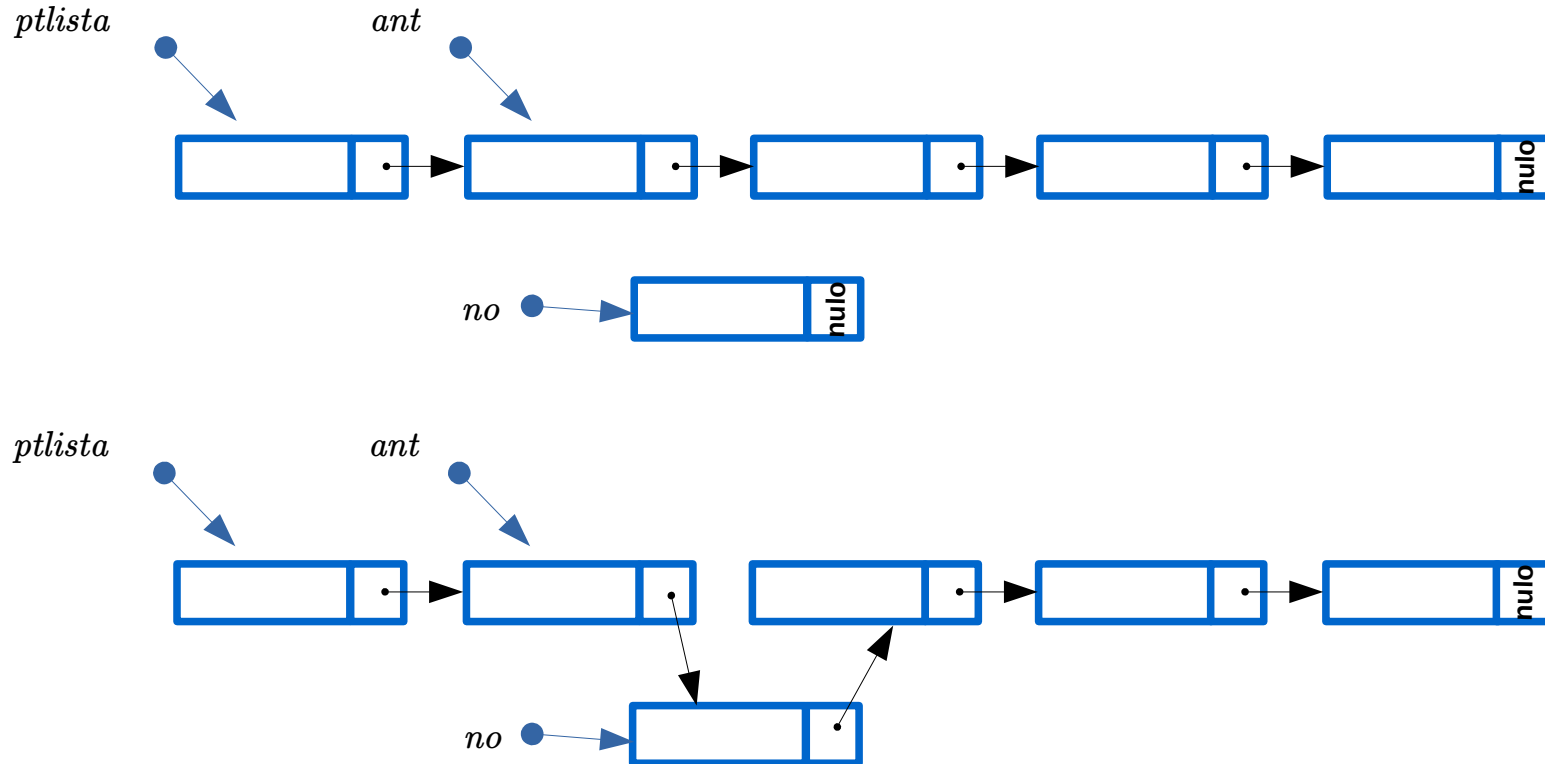
pont := *ptr*

ptr := *nulo*



Lista simplesmente encadeada

- Operações – inserção.



Lista simplesmente encadeada

- Operações – inserção.
- Retorna:
 - -1 no caso de não ser possível inserir (ex: elemento já existe);
 - 0 em caso de sucesso;

função *insere-enc*(*no*)

insere-enc := -1

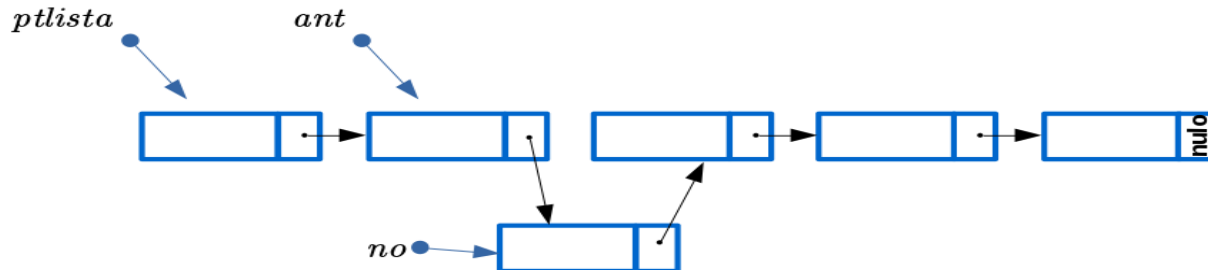
busca-enc(*no.chave*, *ant*, *pont*)

se *pont* = *nulo* então

$no \uparrow .prox := ant \uparrow .prox$

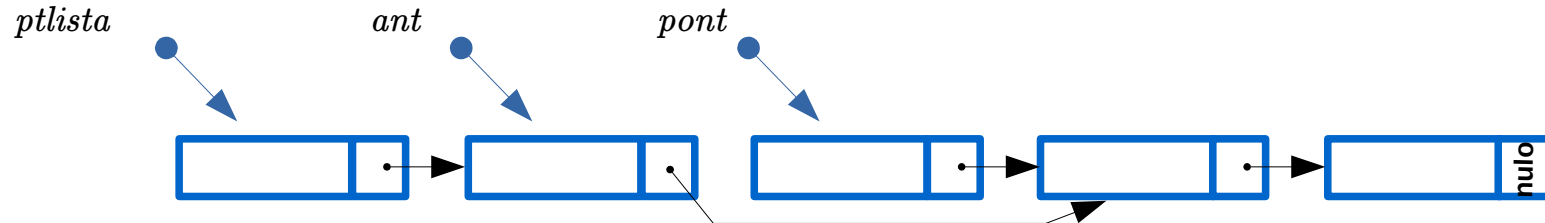
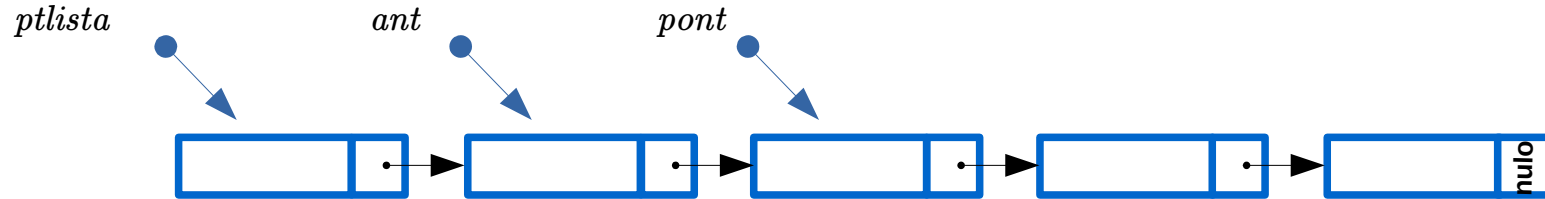
$ant \uparrow .prox := no$

insere-enc := 0



Lista simplesmente encadeada

- Operações – remoção.



Lista simplesmente encadeada

- Operações – remoção.
- Retorna:
 - *nulo* no caso de lista vazia ou elemento inexistente;
 - o *nó* em caso de sucesso;

função *remove-enc*(*x*)

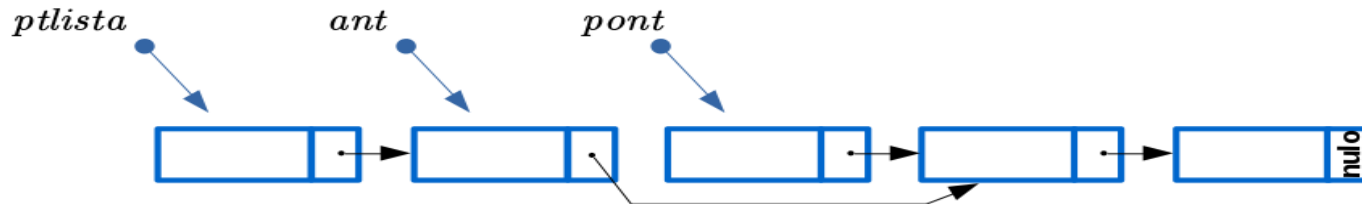
remove-enc := *nulo*

busca-enc(*x*, *ant*, *pont*)

se *pont* ≠ *nulo* então

ant↑.*prox* := *pont*↑.*prox*

remove-enc := *pont*



Referências Bibliográficas

- Estruturas de Dados e Seus Algoritmos. Szwarcfiter J. L.; Markenzon L.. 3a Edição. Editora LTC. 2010.
- Estruturas De Dados Usando C. Tenenbaum A. M.; Langsam Y.; Augenstein M. J.. 1a Edição. Editora Pearson. 1995.
- Introdução a Estruturas de Dados: Com Técnicas de Programação em C. Celes W.; Cerqueira R.; Rangel J.. 2a Edição. Editora Elsevier. 2017.