

Técnicas de programação para Games

Aula02

Bases Matemáticas

Professor Mestre: Adilson Lopes Khouri

14 de outubro de 2018

Sumário

Array x Estruturas Dinâmicas

Cronograma

Aula	Conteúdo
12/04/2018	XP e banco de dados
17/04/2018	Introdução de estruturas de dados
24/04/2018	Arrays / Matrizes e Ordenação
26/04/2018	Recursão
03/05/2018	Lista Ligada
08/05/2018	Pilha, Fila
10/05/2018	Hash
15/05/2018	Árvore Binária
17/05/2018	Heap
22/05/2018	Grafos
24/05/2018	Prova

Array x Estruturas Dinâmicas

- ▶ Arrays são excelentes estruturas de dados, infelizmente tem tamanho fixo. Essa limitação nos obriga a saber o tamanho total a ser alocado previamente ou devemos copiar o array para um array maior sempre que o tamanho estourar.
- ▶ Para evitar esse problema existem estruturas de dados dinâmicas. Onde não há necessidade de alocar memória previamente ou recopiar a estrutura toda para uma nova. Elas crescem a medida que tamanho é solicitado

Lista Ligada - 1 encadeamento

- ▶ A primeira estrutura de dados dinâmica que iremos estudar é a lista ligada com encadeamento simples.
- ▶ Consiste em um conjunto de nós, cada nó armazena dados e um ponteiro com a posição em memória do próximo nó.
- ▶ Cada novo nó a estrutura aloca memória extra para ele e encadeia no restante da lista.
- ▶ No início a lista é vazia com apenas um ponteiro para NULL, ao adicionar o primeiro nó, alocamos memória e fazemos o último nó da lista apontar para o novo nó.

Lista Ligada - 1 encadeamento

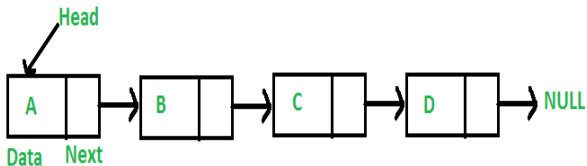


Figura: Lista Ligada - 1 encadeamento

Lista Ligada - 1 encadeamento: Inserção

- ▶ A primeira estrutura de dados dinâmica que iremos estudar é a lista ligada com encadeamento simples.
- ▶ Há três casos para inserção, variando de acordo com a posição do nó na lista:
- ▶ Novo head
- ▶ Entre nós
- ▶ Último nó da lista

Lista Ligada - 1 encadeamento: Inserção

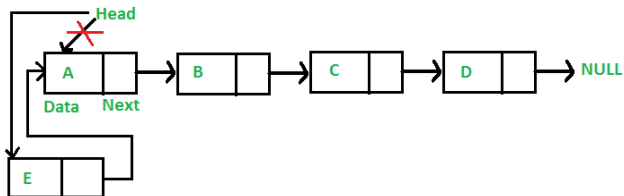


Figura: Lista Ligada - 1 encadeamento: Inserção no início

Lista Ligada - 1 encadeamento: Inserção

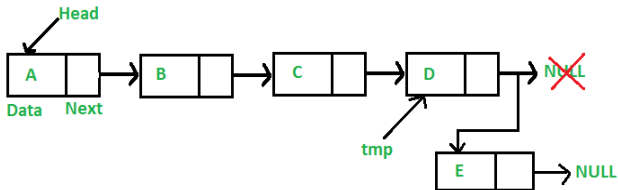


Figura: Lista Ligada - 1 encadeamento: Inserção no fim

Lista Ligada - 1 encadeamento: Inserção

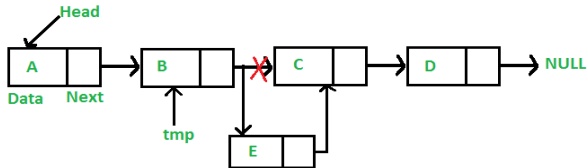


Figura: Lista Ligada - 1 encadeamento: Inserção no meio

Lista Ligada - 1 encadeamento: Deleção

- O anterior aponta para o next do nó deletado

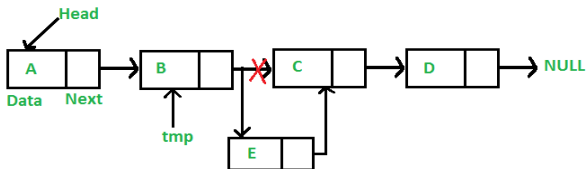


Figura: Lista Ligada - 1 encadeamento: Inserção no meio

Lista Ligada

- ▶ Programar com os alunos em C#
- ▶ Mostrar complexidade

Lista Duplamente Ligada

- ▶ As listas duplamente encadeadas seguem o mesmo raciocínio de listas encadeadas simples... São estruturas dinâmicas que permitem flexibilidade na criação de novos elementos.
- ▶ Possuem como diferencial um encadeamento extra para o nó anterior além do seguinte. Isso permite percorrer com facilidade os nós da lista em ambos os sentidos.

Lista Duplamente Ligada

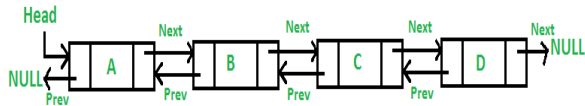


Figura: Lista Ligada - 1 encadeamento: Inserção no meio

Lista Duplamente Ligada: Inserção

- ▶ Há quatro casos para inserção, variando de acordo com a posição do nó na lista:
- ▶ Novo head
- ▶ Após um nó
- ▶ Antes de um nó
- ▶ Último elemento da lista

Lista Duplamente Ligada: Inserção

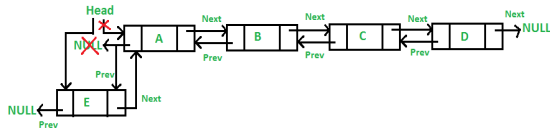


Figura: Lista Ligada - 1 encadeamento: Inserção na frente

Lista Duplamente Ligada: Inserção

► Após "B"

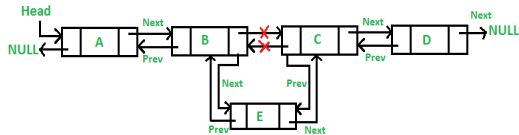


Figura: Lista Ligada - 1 encadeamento: Inserção no meio

Lista Duplamente Ligada: Inserção

► Antes de "C"

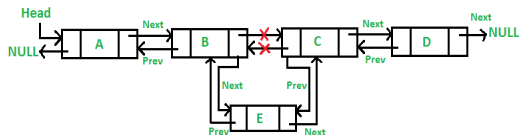


Figura: Lista Ligada - 1 encadeamento: Inserção no meio

Lista Duplamente Ligada: Inserção

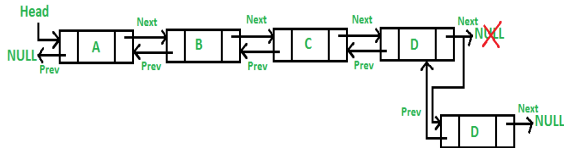


Figura: Lista Ligada - 1 encadeamento: Inserção no fim

Lista Duplamente Ligada: Deleção

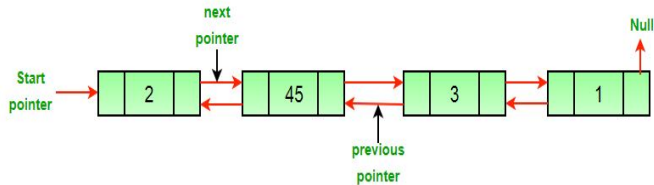


Figura: Lista Ligada - 1 encadeamento: Deleção

Lista Duplamente Ligada: Deleção

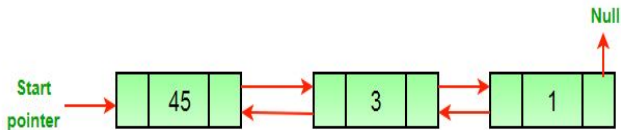


Figura: Lista Ligada - 1 encadeamento: Deleção

Lista Duplamente Ligada: Deleção

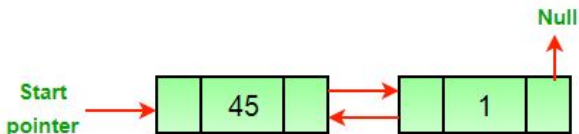


Figura: Lista Ligada - 1 encadeamento: Deleção

Lista Duplamente Ligada: Deleção

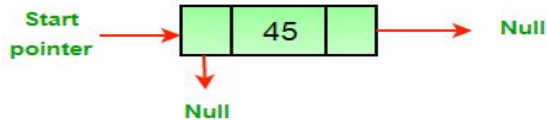


Figura: Lista Ligada - 1 encadeamento: Deleção

Lista Duplamente Ligada

- ▶ Programar com os alunos em C#
- ▶ Mostrar complexidade

Lista Circular

- ▶ Listas ligadas podem ser circulares, onde o último nó aponta para o primeiro ao invés de apontar para: "null"
- ▶ Normalmente usadas para implementar filas
- ▶ Usadas pelo OS para alocar recursos por um período de tempo

Lista Circular

- ▶ O nó, operações de deleção e inserção são como no caso da lista simples usando os nós de head e tail para gerenciar a lista.

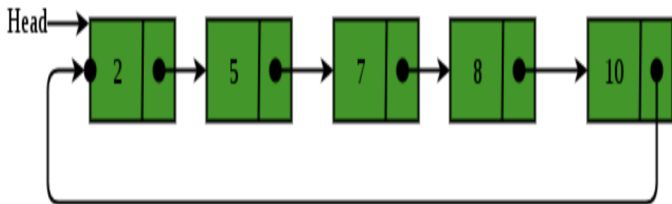


Figura: Lista Circular

Lista Circular

- ▶ Programar com os alunos em C#
- ▶ Mostrar complexidade

Dúvidas...

Alguma dúvida?

Contato

- ▶ E-mail: *0800dirso@gmail.com* (alunos SENAC)
- ▶ E-mail: *adilson.khoury.usp@gmail.com*
- ▶ Phone: +55119444 – 26191
- ▶ [Linkedin](#)
- ▶ [Lattes](#)
- ▶ [GitHub](#)

Referências

- [1] A. V. Aho, J. E. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983.
- [2] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.
- [3] Beck, *Test Driven Development: By Example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [4] M. Fowler, *Refactoring: Improving the Design of Existing Code*. Boston, MA, USA: Addison-Wesley, 1999.
- [5] Geeks. (2018) A computer science portal for geeks. [Online]. Available: <https://www.geeksforgeeks.org>