

Listas Ligadas

Vetores

- permitem a implementação fácil de listas de valores de forma sequencial
- estão alocados contiguamente na memória
 - pode ser que tenhamos espaço na memória
 - mas não para alocar um vetor do tamanho desejado
- têm um tamanho fixo
 - ou alocamos um vetor pequeno e o espaço pode acabar
 - ou alocamos um vetor grande e desperdiçamos memória

Como funcionam os vetores?

- Para algumas tarefas eles podem não ser eficientes (justamente pelo fato de ser alocação contínua):
 - Adicionar um elemento na primeira posição de um Vetor, por exemplo, consome muito tempo, pois temos de deslocar todos os outros elementos uma posição para a frente
 - isso piora muito conforme o número de elementos no vetor cresce
 - Analogamente, remover um elemento da primeira posição implica em deslocar todos os outros elementos que estão na sua frente para trás.

Uma alternativa

- Uma implementação de Lista na qual as operações de adicionar e remover um elemento na primeira posição seja computacionalmente eficiente.
- Isso é possível utilizando-se

Listas Ligadas

- Estrutura que permite a inserção e a remoção de elementos nos “extremos” da lista de forma eficiente, computacionalmente falando

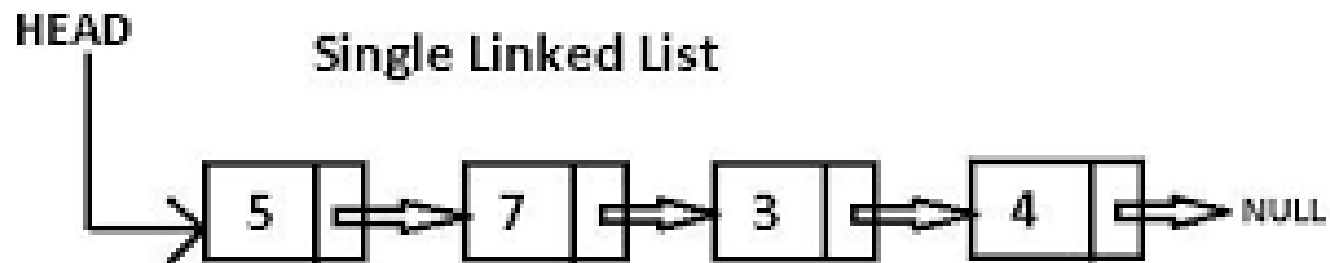
Listas Ligadas

- Nó: elemento alocado dinamicamente que contém
 - um conjunto de dados
 - um ponteiro para outro nó
- Lista ligada:
 - Conjunto de nós ligados entre si de maneira lógica e sequencial

Listas Ligadas

- Tipicamente, em uma lista ligada há um ou dois pontos conhecidos de acesso
 - o **primeiro** elemento e, eventualmente, o **último**
- Desse modo, o armazenamento de uma lista não requer uma área contígua de memória
- A lista ligada é acessada a partir de uma variável de referência (um ponteiro)
- um ponteiro pode estar vazio (aponta para null)

Tipos



BORA IMPLEMENTAR!!!