Algoritmos e Programação I

Módulo 4 - Lista Estática

Profª. Elisa de Cássia Silva Rodrigues

Lista

- Definição:
  - Estrutura de dados utilizada para armazenar e organizar uma sequência de elementos do mesmo tipo.

- Características:
  - Seus elementos possuem estrutura interna abstraída.
  - Uma lista pode possuir elementos repetidos, ser ordenada ou não.
  - Uma lista pode possuir $N >= 0$ elementos ou itens.
  - Se $N = 0$, dizemos que a lista é vazia.

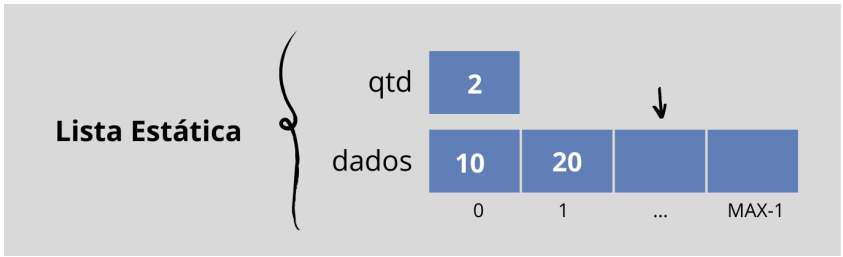
A implementação de operações de uma lista depende do tipo de alocação de memória usada (estática ou dinâmica).

Alocação de Memória

- Definição:
  - Processo de reserva de memória para armazenamento de dados durante a execução de um programa.
- Alocação estática:
  - A memória é alocada automaticamente no momento da compilação.
  - Vantagens:
    - ★ Os dados ficam armazenados sequencialmente na memória (vetores).
    - ★ Programador não precisa se preocupar em gerenciar a memória.
  - Desvantagens:
    - ★ Programador não tem controle sob o tempo de vida das variáveis.
    - ★ Quantidade de memória utilizada DEVE ser definida previamente.
    - ★ Espaço reservado não pode ser alterado.
    - ★ Podem haver espaços reservados desnecessariamente.

Lista Estática

- Definição:
  - Estrutura de dados do tipo lista que é definida utilizando alocação estática e acesso sequencial dos elementos.
- Características:
  - Tipo mais simples de lista.
  - Definida por um vetor com elementos sequenciais na memória.
  - Exige a definição prévia do número máximo de elementos da lista.
  - Campo adicional para armazenar a quantidade de elementos na lista.



Lista Estática

- Vantagens:
  - Acesso rápido e direto aos elementos (índice do vetor).
  - Tempo constante para acessar um elemento.
  - Facilidade para modificar as suas informações.
- Desvantagens:
  - Definição prévia do tamanho do vetor e, consequentemente, da lista.
  - Dificuldade para inserir e remover um elemento entre outros dois:
    - ★ É necessário deslocar os elementos para abrir espaço dentro do vetor.

Lista Estática

- Definição do TAD Lista Estática:
  - Definir os arquivos `listaEstatica.h` e `listaEstatica.c`.
  - Declarar o tipo de dado que irá representar a lista no arquivo `.h`:

```
typedef struct lista Lista;
```

  - Definir o tipo de dado que será armazenado dentro da lista (`int`).
  - Declarar a estrutura para representar a lista estática no arquivo `.c`:

```
struct lista{
  int qtd;
  int dados[MAX]; // MAX representa o tamanho da lista
};
```

- Declarar um ponteiro do tipo `Lista` para acessar o TAD (`main.c`):

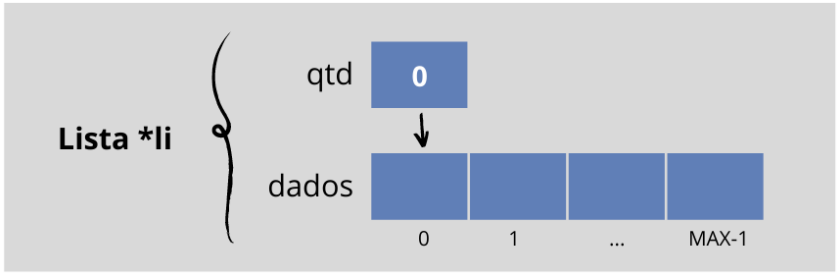
```
Lista *li;
```

Lista Estática

- Definição das operações do TAD Lista Estática:
  - Declaração dos protótipos das funções no arquivo `.h`.
  - Implementação das funções no arquivo `.c`.
- Operações básicas:
  - Criação da lista.
  - Inserção de um elemento na lista.
  - Remoção de um elemento da lista.
  - Busca por um elemento da lista.
  - Destruição da lista.
  - Informações sobre tamanho da lista.
  - Informação sobre a lista estar vazia ou cheia.

Lista Estática

- Criação da lista:
  - Antes de usar uma lista é preciso criar uma lista vazia.
  - Isto é, alocar um espaço na memória para a estrutura:
    - ★ Alocação dinâmica da estrutura `Lista` usando `malloc()`.
  - A lista está vazia, quando `qtd = 0`.



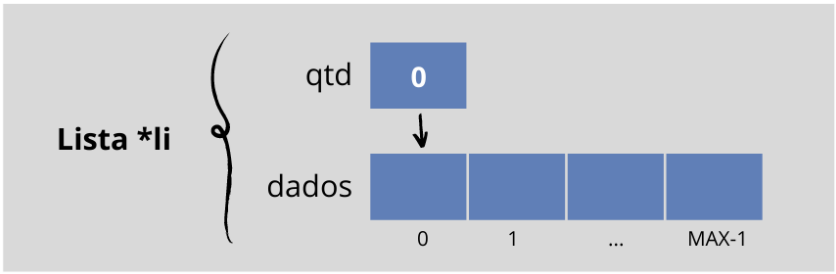
Note que o vetor `dados[]` que armazena os elementos da lista é alocado estaticamente durante a alocação da estrutura `Lista`.

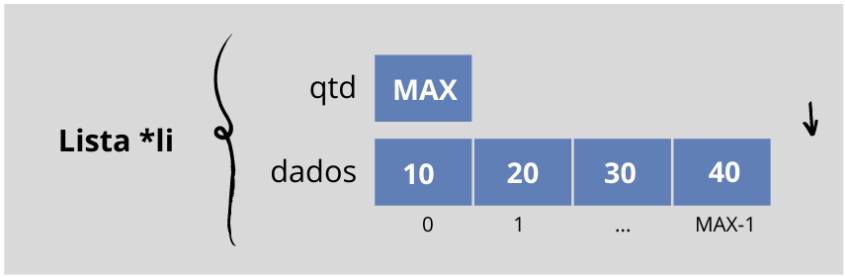
Lista Estática

- Destruição da lista:
  - Deve-se liberar a memória alocada para a estrutura:
    - ★ Liberação da estrutura `Lista` usando `free()`.

Lista Estática

- Informações básicas sobre a lista:
  - Tamanho da lista (valor do campo `qtd`).
  - Lista vazia (`qtd = 0`).
  - Lista cheia (`qtd = MAX`).



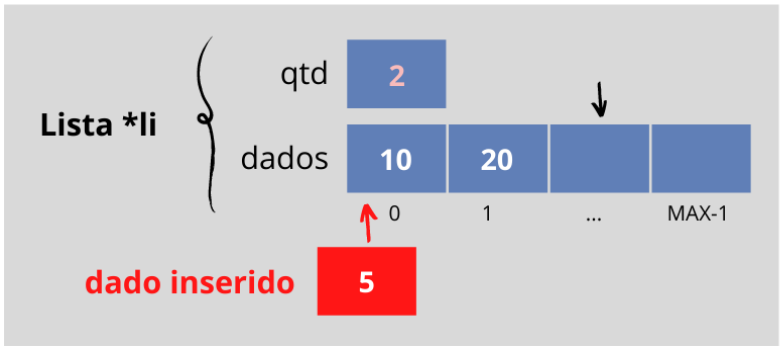


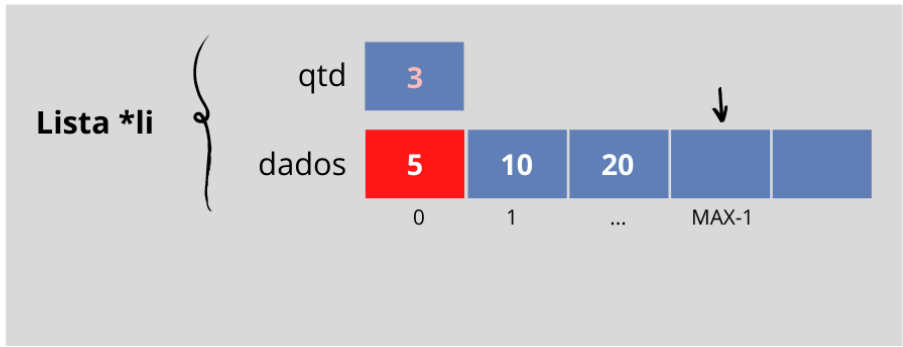
Lista Estática

- Inserção:
  - Ato de guardar elementos dentro da lista.
  - Tipos de inserção:
    - ★ No início da lista.
    - ★ No meio da lista (usada em listas ordenadas).
    - ★ No final da lista.
  - Operação de inserção envolve o teste de estouro da lista:
    - ★ Necessário verificar se é possível inserir um novo elemento na lista.
    - ★ Ou seja, se a lista não está cheia.

Lista Estática

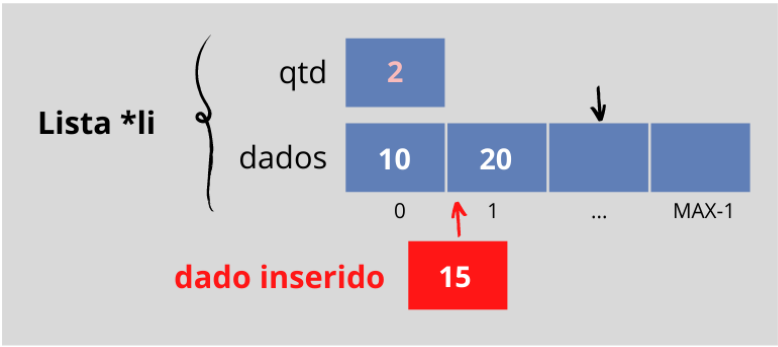
- Inserção no início da lista:
  - Envolve o deslocamento de todos os elementos do vetor.
  - Devem ser deslocados uma posição para frente.
  - Então, o novo elemento é inserido na posição zero do vetor.
  - Ao fim da operação deve-se incrementar o campo `qtd`.

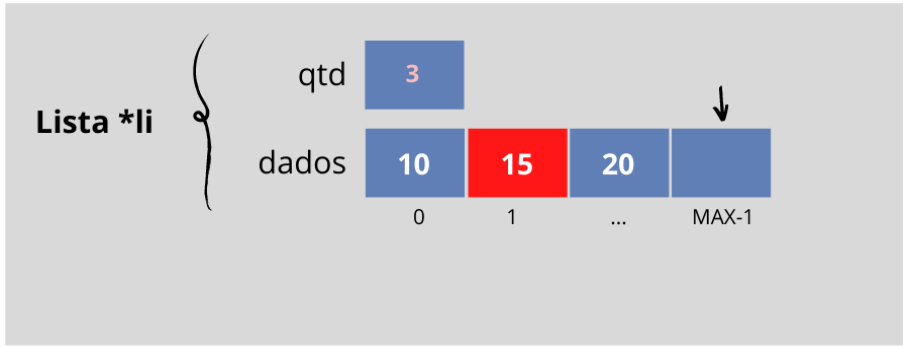




Lista Estática

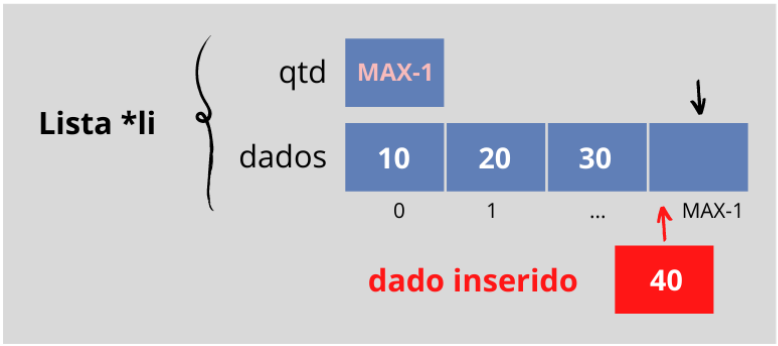
- Inserção no meio da lista:
  - Utilizada para inserir um elemento de forma ordenada na lista.
  - Envolve o deslocamento de todos os elementos do vetor a partir da posição onde o elemento será inserido.
  - Ao fim da operação deve-se incrementar o campo `qtd`.





Lista Estática

- Inserção no final da lista:
  - Não envolve o deslocamento de elementos do vetor.
  - O novo elemento é inserindo logo após a última posição ocupada do vetor, ou seja, na primeira posição livre.
  - Ao fim da operação deve-se incrementar o campo `qtd`.

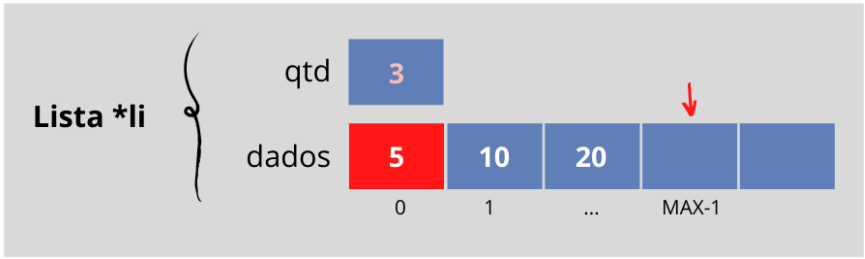


Lista Estática

- Remoção:
  - Existindo uma lista, e ela possuindo elementos, é possível excluí-los.
  - Tipos de remoção:
    - ★ No início da lista.
    - ★ No meio da lista (usada para remover um elemento específico).
    - ★ No final da lista.
  - Operação de remoção envolve o teste de lista vazia.
    - ★ Necessário verificar se existem elementos dentro da lista.
    - ★ Ou seja, se a lista não está vazia.

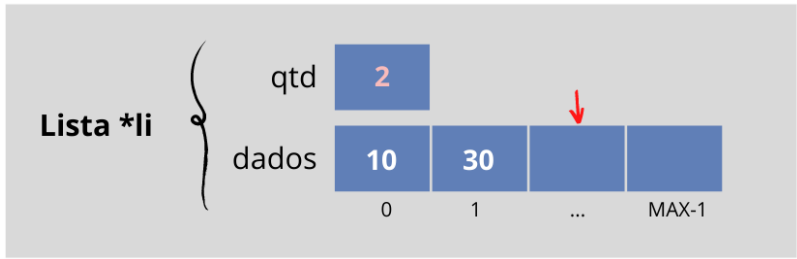
Lista Estática

- Remoção do início da lista:
  - Envolve o deslocamento de todos os elementos do vetor.
  - Devem ser deslocados uma posição para trás.
  - Ao fim da operação deve-se decrementar o campo `qtd`.



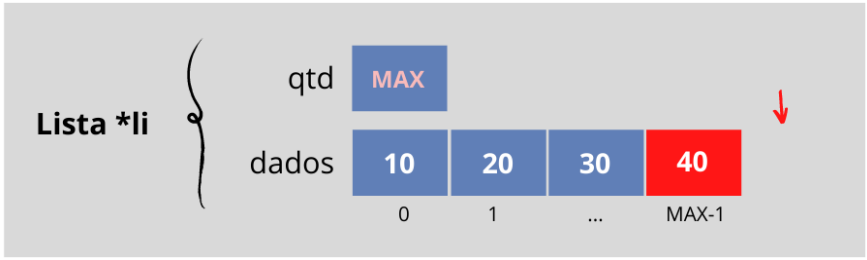
Lista Estática

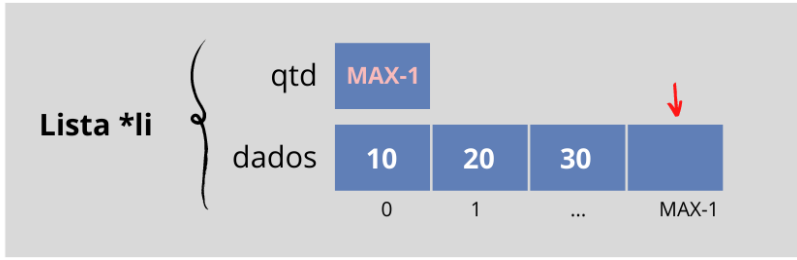
- Remoção do meio da lista:
  - Utilizada para remover um elemento específico da lista.
  - Antes da remoção é preciso buscar o elemento na lista.
  - Envolve o deslocamento de todos os elementos do vetor a partir da posição onde o elemento será removido.
  - Ao fim da operação deve-se decrementar o campo `qtd`.



Lista Estática

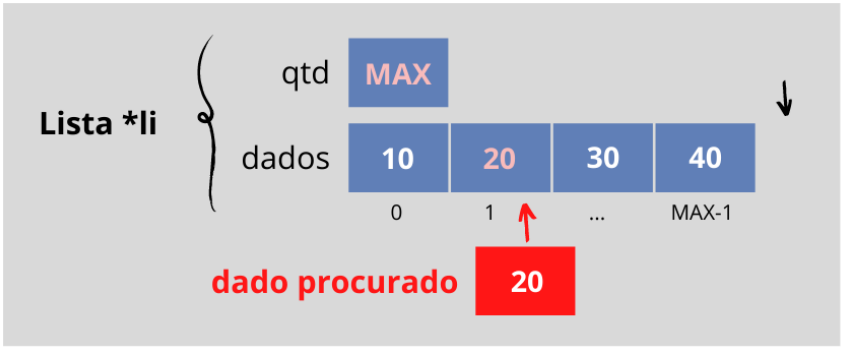
- Remoção do final da lista:
  - Não envolve o deslocamento de elementos do vetor.
  - O elemento da última posição ocupada do vetor é removido.
  - Ao fim da operação deve-se decrementar o campo `qtd`.

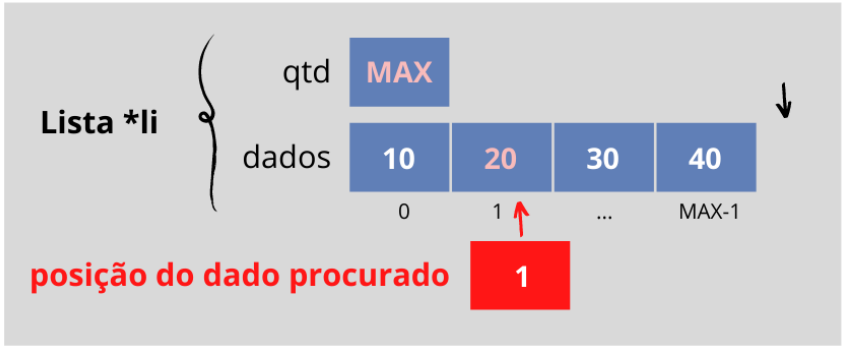




Lista Estática

- Busca:
  - Envolve percorrer a lista à procura do elemento procurado.
  - Essa operação pode ocorrer de duas formas:
    - ★ Dado um elemento, devolve a posição dele na lista.
    - ★ Dada uma posição, devolve o elemento daquela posição.





Lista Estática

- Quando usar esse tipo de lista?
  - Em aplicações com listas pequenas.
    - ★ Porque a alocação de memória é sequencial.
  - Quando o tamanho máximo da lista é bem definido.
    - ★ Porque é necessário definir previamente o tamanho do vetor.
  - Quando ocorrem inserções e remoções apenas no final da lista.
    - ★ Porque não é necessário deslocar elementos do vetor.
  - Quando a operação de busca é mais frequente.
    - ★ Porque o tempo para acessar um elemento é constante.

Implementação:

https://replit.com/@elisa_rodrigues/Modulo4-ListaEstatica

Referências Bibliográficas

1. BACKES, A. ***Estrutura de dados descomplicada em linguagem C***. 2016.

   -> **Capítulo 5: Listas**

   -> **Material Complementar - Vídeo aulas (3ª a 9ª):**

   https://programacaodescomplicada.wordpress.com/indice/estrutura-de-dados/