

# *Algoritmos e Estruturas de dados*

*Listas encadeadas estáticas*

Prof. Dr. Fábio Rodrigues de la Rocha

## *Listas estáticas x Listas estáticas encadeadas*

Anteriormente foi estudado o assunto de listas estáticas que são armazenadas como vetores. Ou seja, tem seu tamanho sempre fixo. Sobre estas listas foram realizadas operações de consulta a elemento, pesquisa a elemento e inserção de elementos. Vimos que as operações de inserção poderiam ser de 2 tipos: Inserção desordenada e ordenada. Para o caso da inserção ordenada, vimos que era necessário realizar deslocamentos de elementos para mantê-los em ordem.

## *Listas estáticas x Listas estáticas encadeadas*

Agora, veremos como realizar inserções numa estrutura de dados onde os elementos podem estar fisicamente desordenados mas estarão logicamente ordenados.

## *Listas estáticas x Listas estáticas encadeadas*

Agora, veremos como realizar inserções numa estrutura de dados onde os elementos podem estar fisicamente desordenados mas estarão logicamente ordenados.

Estado original



 Em uso

 Livre

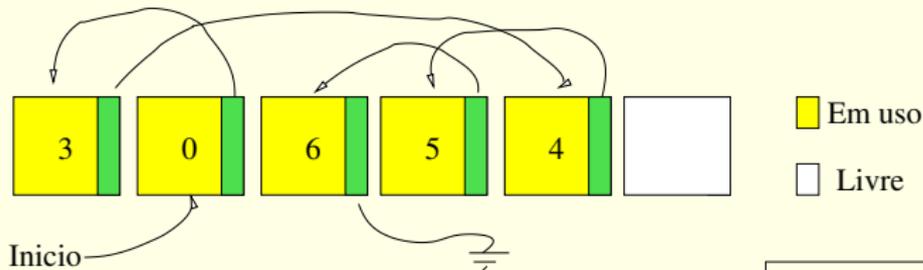
Depois de inserir o elemento 2



Foi necessario movimentar 4 elementos!

# Listas

Estado original

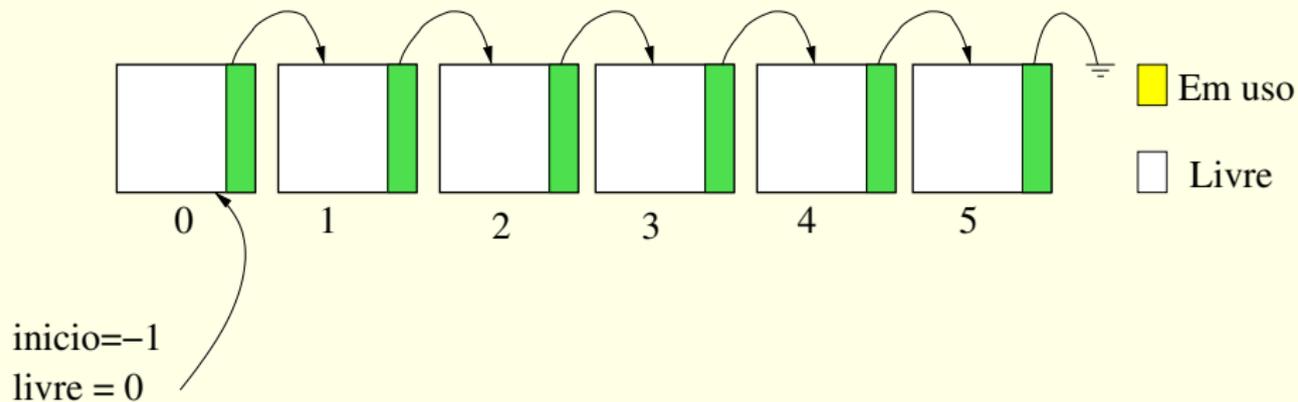


```
typedef struct {  
    int valor;  
    int prox;  
}Tipo_Bloco;
```

```
typedef struct {  
    Tipo_Bloco vet[60];  
    int inicio; int livre;  
}Tipo_Lista_estatica_encadeada;
```

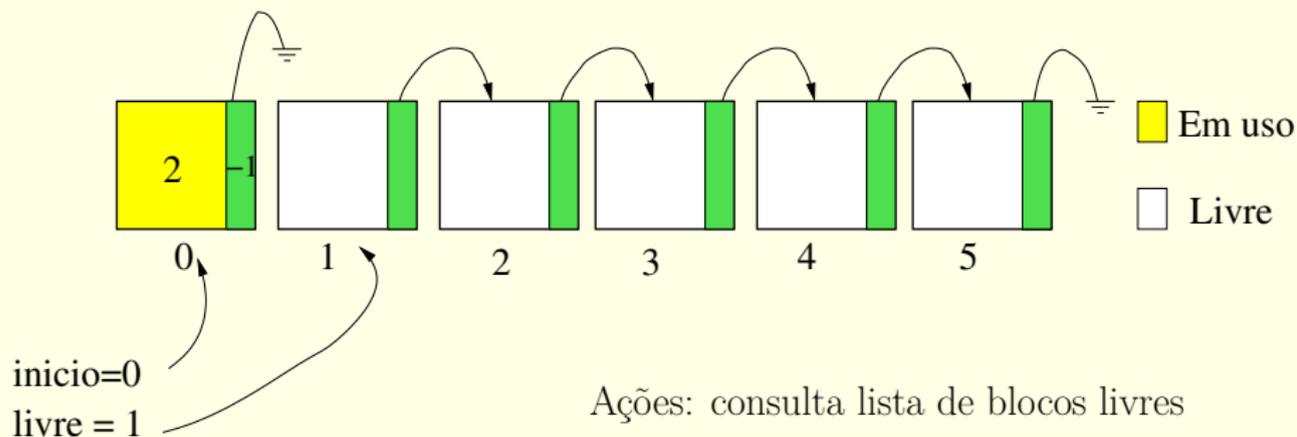
# Listas

Operação inicializa



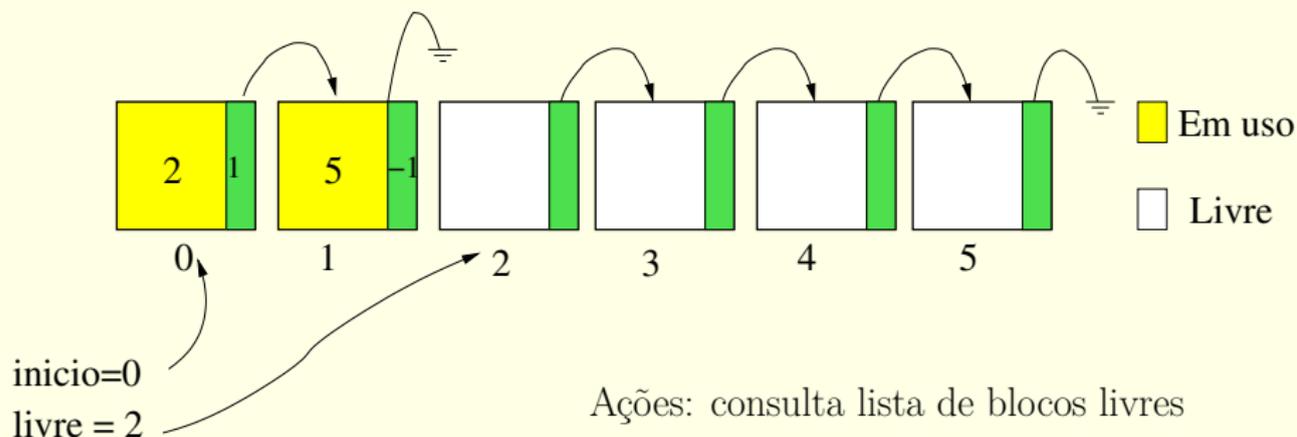
# Listas

Operação insere: valor 2



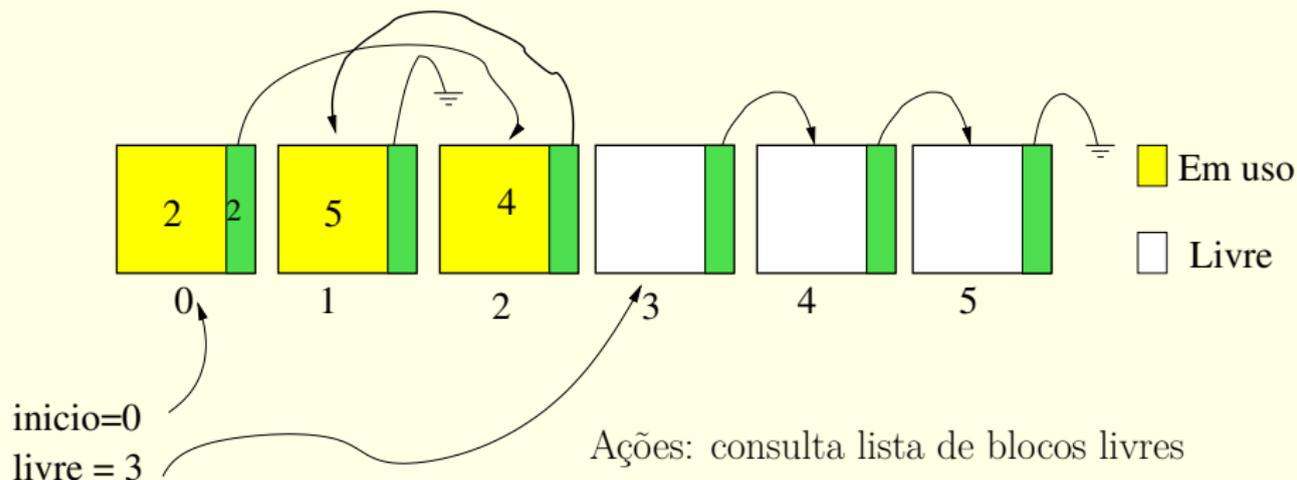
# Listas

Operação insere: valor 5



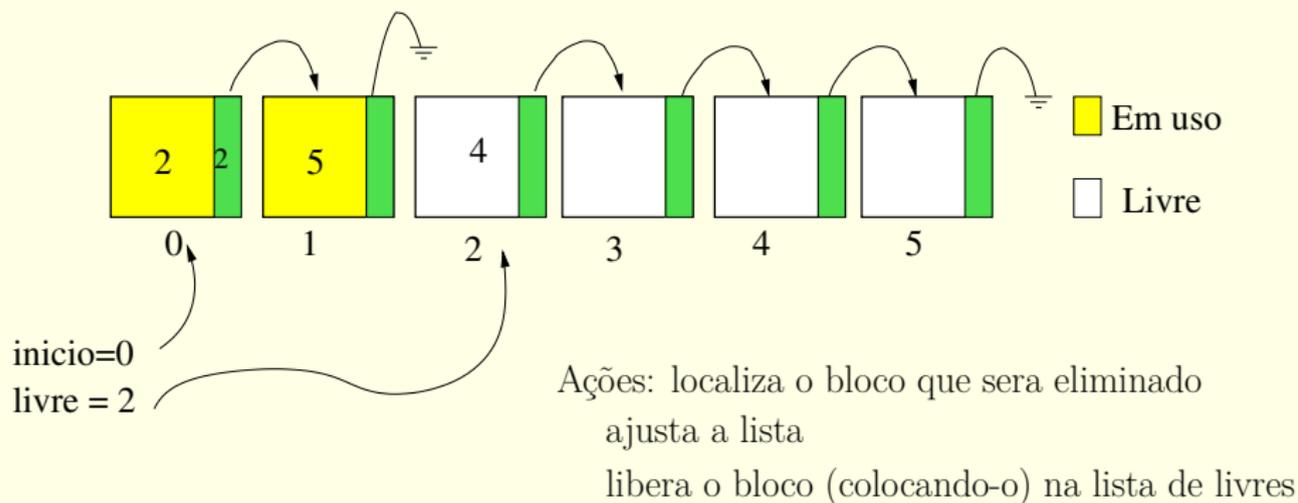
# Listas

Operação insere: valor 4



# Listas

Operação remove o valor 4



Implemente em C as rotinas `insere_ordenado()`, `pesquisa()`, `inicializa()`, `remove()` e `mostra()`