

Listas

Estruturas de Dados e Algoritmos



Prof. Daniel Saad Nogueira
Nunes

IFB – Instituto Federal de Brasília,
Campus Taguatinga



Sumário

- 1 Introdução
- 2 Listas Encadeadas
- 3 Listas Duplamente Encadeadas
- 4 Exemplos



Sumário

1 Introdução



Tipo Abstrato de Dados

TAD

- Um tipo abstrato de dado (TAD) é um modelo matemático para uma classe de estruturas de dados que possuem uma semântica similar.
- Um TAD define as operações essenciais sobre uma estrutura de dados.



Listas

- Lista é um TAD definido como uma sequência de valores em que um determinado valor pode ocorrer múltiplas vezes.
- A lista possui uma cabeça (primeiro elemento da sequência) e uma cauda (último elemento da sequência).
- É interessante que listas possuam operações eficientes na cabeça e na cauda.



Listas





Listas

Operações sobre Listas

- Verificar se a lista está vazia;
- Inserção de qualquer posição da lista;
- Inserção na cabeça;
- Inserção na cauda;
- Remoção de qualquer posição da lista;
- Remoção da cabeça da lista;
- Remoção da cauda da lista;
- Acesso à cabeça da lista;
- Acesso à cauda da lista;
- Acesso à qualquer posição da lista;



Listas

- Listas podem ser implementadas por vetores ou estruturas dinâmicas auto-referenciadas.
- Nosso foco será em estruturas auto-referenciadas.



Listas Encadeadas

- Uma estrutura auto-referenciado é aquela que contém uma referencia para um elemento do mesmo tipo.
- Em C, isto é alcançado através de **ponteiros**.
- Listas Auto-Referenciadas, Listas Encadeadas ou Listas Ligadas!



Listas Encadeadas

- Em comparação com a implementação em vetores, listas encadeadas possuem vantagens e desvantagens.



Listas Encadeadas

Vantagens

- Estrutura dinâmica: pode aumentar facilmente.
- Inserção em qualquer posição da lista não ocasiona um deslocamento dos elementos posteriores.
- Permite utilizar regiões não contíguas de memória.
- Gerência de simples.



Listas Encadeadas

Desvantagens

- Espaço extra para armazenar ponteiros (implícitos em vetores).
- Não possui acesso aleatório em tempo constante.



Sumário

2 Listas Encadeadas



Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Encadeadas: Definição

```
7  /**
8   @brief list_node_t Definição de nó de lista ligada.
9   O nó de lista ligada contém um ponteiro para um dado genérico (data)
10  e um ponteiro para o próximo nó da lista.
11  **/
12  typedef struct list_node_t {
13      int data; /*Dado da lista*/
14      struct list_node_t *next; /*ponteiro para o próximo elemento*/
15  } list_node_t;
```



Listas Encadeadas: Definição

```
17  /**
18   @brief list_t Definição do tipo lista. Contém ponteiros para a cabeça e cauda
19   da lista e o tamanho da lista.
20  **/
21  typedef struct list_t {
22      list_node_t *head; /*Cabeça da Lista*/
23      list_node_t *tail; /*Cauda da Lista*/
24      size_t size;      /*tamanho da lista*/
25  } list_t;
```




Sumário

2 Listas Encadeadas

- Definição
- **Inicialização**
- Funções auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Encadeadas: Inicialização





Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Encadeadas: Funções Auxiliares

Retorna o tamanho da lista.





Listas Encadeadas: Funções Auxiliares

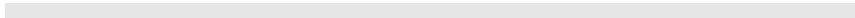
Retorna verdadeiro se e somente se a lista está vazia.





Listas Encadeadas: Funções Auxiliares

Cria um novo nó e o inicializa com um valor.





Listas Encadeadas: Funções Auxiliares

Remove um nó da memória.





Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- **Inserção**
- Remoção
- Acesso
- Limpeza
- Análise



Listas Encadeadas: Inserção

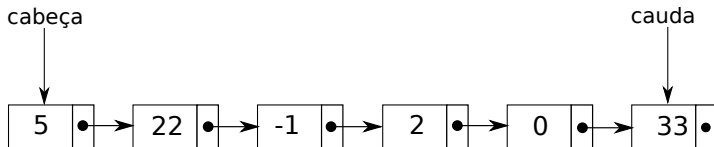
Inserção

- Inserções na cabeça e na cauda da lista, podem ser efetuadas em $\Theta(1)$.
- Inserções em posições aleatórias, requerem acesso sequencial na lista, e portanto tempo $\Theta(n)$.



Listas Encadeadas: Inserção na Cabeça

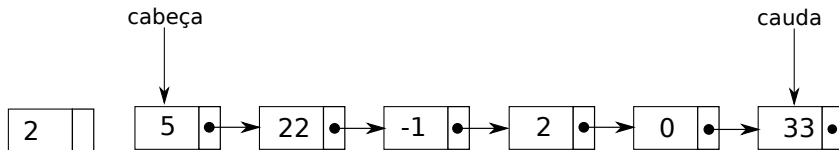
Para inserir na cabeça da lista, criamos um novo nó que aponta para a cabeça atual e depois movemos o ponteiro da cabeça para o novo nó. Se a lista estava vazia, a cauda também deve apontar para o nó recém inserido.





Listas Encadeadas: Inserção na Cabeça

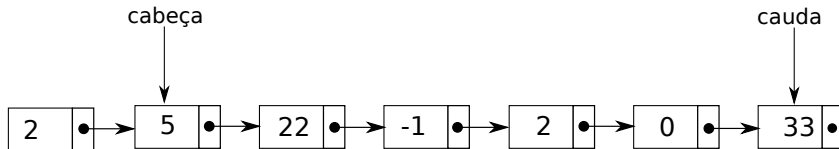
Para inserir na cabeça da lista, criamos um novo nó que aponta para a cabeça atual e depois movemos o ponteiro da cabeça para o novo nó. Se a lista estava vazia, a cauda também deve apontar para o nó recém inserido.





Listas Encadeadas: Inserção na Cabeça

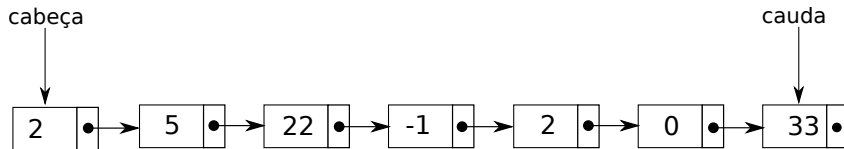
Para inserir na cabeça da lista, criamos um novo nó que aponta para a cabeça atual e depois movemos o ponteiro da cabeça para o novo nó. Se a lista estava vazia, a cauda também deve apontar para o nó recém inserido.





Listas Encadeadas: Inserção na Cabeça

Para inserir na cabeça da lista, criamos um novo nó que aponta para a cabeça atual e depois movemos o ponteiro da cabeça para o novo nó. Se a lista estava vazia, a cauda também deve apontar para o nó recém inserido.





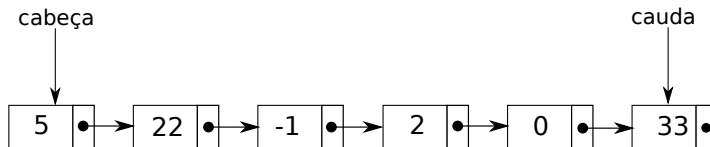
Listas Encadeadas: Inserção na Cabeça





Listas Encadeadas: Inserção na Cauda

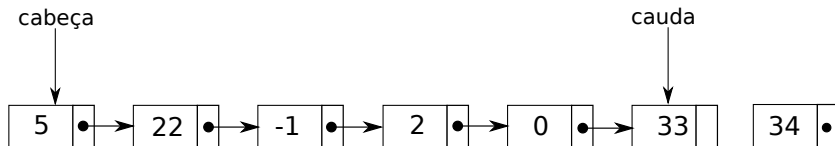
Para inserir um nó na cauda, basta criar um novo nó e fazer com que a cauda aponte para ele. Após isso, a cauda passa a apontar para o nó criado. Caso a lista estivesse vazia, a cabeça também deve apontar para o nó criado





Listas Encadeadas: Inserção na Cauda

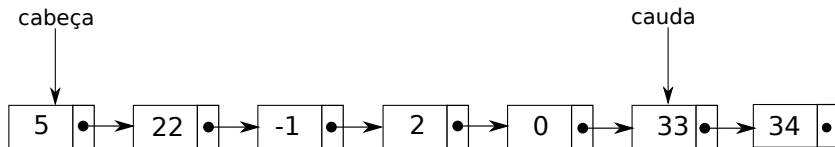
Para inserir um nó na cauda, basta criar um novo nó e fazer com que a cauda aponte para ele. Após isso, a cauda passa a apontar para o nó criado. Caso a lista estivesse vazia, a cabeça também deve apontar para o nó criado





Listas Encadeadas: Inserção na Cauda

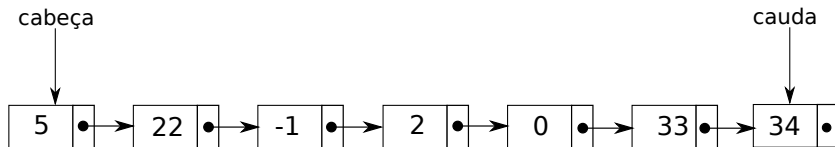
Para inserir um nó na cauda, basta criar um novo nó e fazer com que a cauda aponte para ele. Após isso, a cauda passa a apontar para o nó criado. Caso a lista estivesse vazia, a cabeça também deve apontar para o nó criado





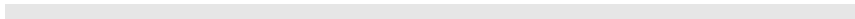
Listas Encadeadas: Inserção na Cauda

Para inserir um nó na cauda, basta criar um novo nó e fazer com que a cauda aponte para ele. Após isso, a cauda passa a apontar para o nó criado. Caso a lista estivesse vazia, a cabeça também deve apontar para o nó criado





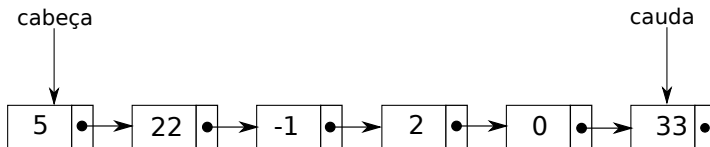
Listas Encadeadas: Inserção na Cauda





Listas Encadeadas: Inserção

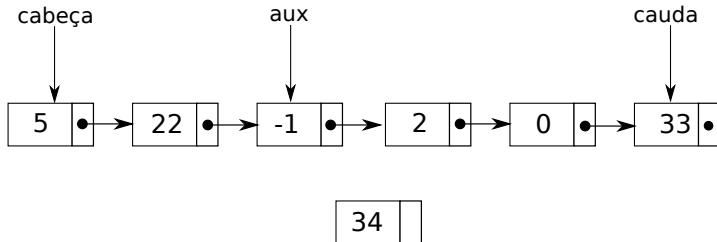
Para inserir em uma posição arbitrária, precisamos percorrer a list até o elemento que antecede a posição de inserção. O novo nó passa a apontar para o nó que sucede este elemento e o elemento passa a apontar para o novo nó.





Listas Encadeadas: Inserção

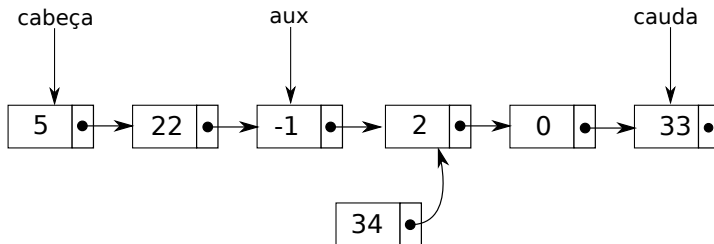
Para inserir em uma posição arbitrária, precisamos percorrer a list até o elemento que antecede a posição de inserção. O novo nó passa a apontar para o nó que sucede este elemento e o elemento passa a apontar para o novo nó.





Listas Encadeadas: Inserção

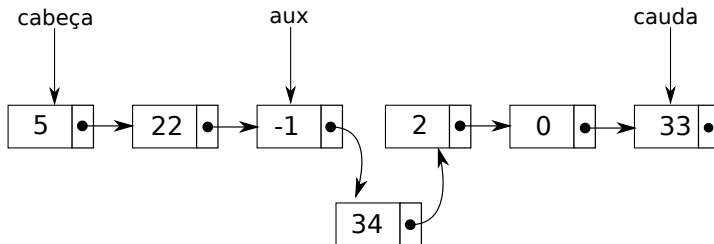
Para inserir em uma posição arbitrária, precisamos percorrer a list até o elemento que antecede a posição de inserção. O novo nó passa a apontar para o nó que sucede este elemento e o elemento passa a apontar para o novo nó.





Listas Encadeadas: Inserção

Para inserir em uma posição arbitrária, precisamos percorrer a list até o elemento que antecede a posição de inserção. O novo nó passa a apontar para o nó que sucede este elemento e o elemento passa a apontar para o novo nó.





Listas Encadeadas: Inserção





Listas Encadeadas: Inserção





Sumário

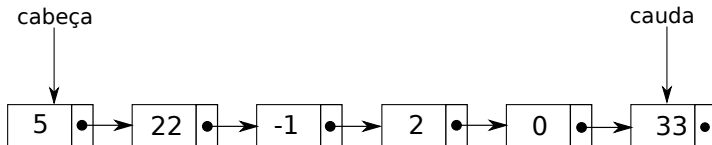
2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- **Remoção**
- Acesso
- Limpeza
- Análise



Listas Encadeadas: Remoção na Cabeça

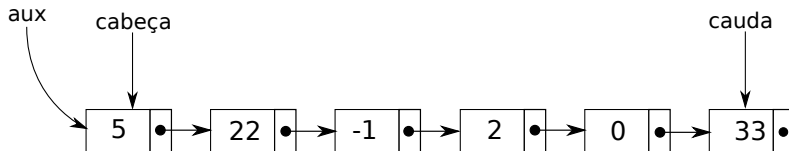
Para remover a cabeça da lista, utilizamos um ponteiro auxiliar que apontará para a cabeça, atualizamos a cabeça para o próximo elemento e removemos o nó apontado pelo ponteiro auxiliar. Caso a lista fique vazia após a remoção deste nó, a cauda deve apontar para NULL.





Listas Encadeadas: Remoção na Cabeça

Para remover a cabeça da lista, utilizamos um ponteiro auxiliar que apontará para a cabeça, atualizamos a cabeça para o próximo elemento e removemos o nó apontado pelo ponteiro auxiliar. Caso a lista fique vazia após a remoção deste nó, a cauda deve apontar para NULL.





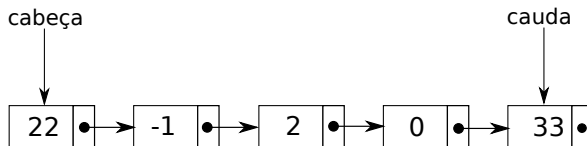
Listas Encadeadas: Remoção na Cabeça

Para remover a cabeça da lista, utilizamos um ponteiro auxiliar que apontará para a cabeça, atualizamos a cabeça para o próximo elemento e removemos o nó apontado pelo ponteiro auxiliar. Caso a lista fique vazia após a remoção deste nó, a cauda deve apontar para NULL.



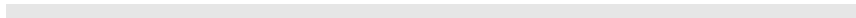
Listas Encadeadas: Remoção na Cabeça

Para remover a cabeça da lista, utilizamos um ponteiro auxiliar que apontará para a cabeça, atualizamos a cabeça para o próximo elemento e removemos o nó apontado pelo ponteiro auxiliar. Caso a lista fique vazia após a remoção deste nó, a cauda deve apontar para NULL.





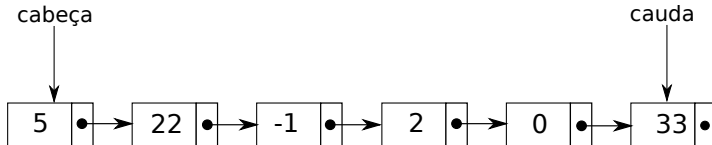
Listas Encadeadas: Remoção na Cabeça





Listas Encadeadas: Remoção na Cauda

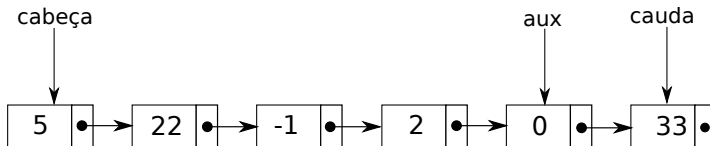
Para remover a cauda, temos que percorrer a lista com um ponteiro auxiliar até chegar ao penúltimo elemento. Assim podemos remover a cauda, fazer com que o nó apontado pelo ponteiro auxiliar aponte para NULL e atualizar a cauda. Caso a lista fique vazia, também deve-se atualizar a cabeça para NULL.





Listas Encadeadas: Remoção na Cauda

Para remover a cauda, temos que percorrer a lista com um ponteiro auxiliar até chegar ao penúltimo elemento. Assim podemos remover a cauda, fazer com que o nó apontado pelo ponteiro auxiliar aponte para NULL e atualizar a cauda. Caso a lista fique vazia, também deve-se atualizar a cabeça para NULL.





Listas Encadeadas: Remoção na Cauda

Para remover a cauda, temos que percorrer a lista com um ponteiro auxiliar até chegar ao penúltimo elemento. Assim podemos remover a cauda, fazer com que o nó apontado pelo ponteiro auxiliar aponte para NULL e atualizar a cauda. Caso a lista fique vazia, também deve-se atualizar a cabeça para NULL.

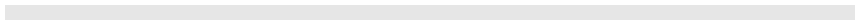


Listas Encadeadas: Remoção na Cauda

Para remover a cauda, temos que percorrer a lista com um ponteiro auxiliar até chegar ao penúltimo elemento. Assim podemos remover a cauda, fazer com que o nó apontado pelo ponteiro auxiliar aponte para NULL e atualizar a cauda. Caso a lista fique vazia, também deve-se atualizar a cabeça para NULL.

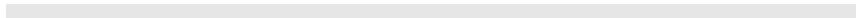


Listas Encadeadas: Remoção na Cauda





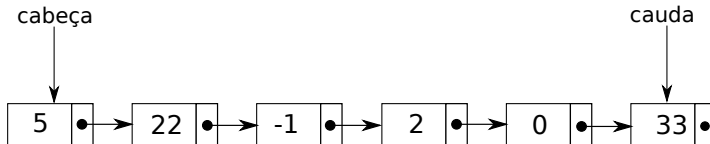
Listas Encadeadas: Remoção na Cauda





Listas Encadeadas: Remoção

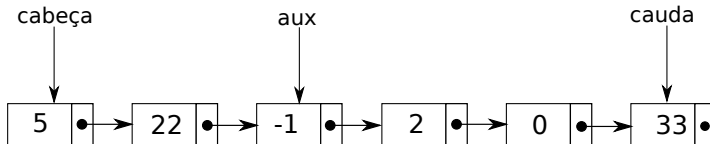
Para remover um elemento arbitrário, devemos percorrer a lista até o nó anterior a posição de remoção para que ele aponte para o nó que sucede o nó a ser removido. Em seguida, removemos o nó.





Listas Encadeadas: Remoção

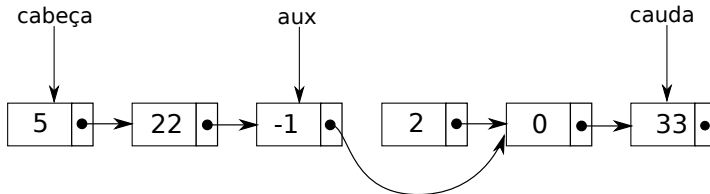
Para remover um elemento arbitrário, devemos percorrer a lista até o nó anterior a posição de remoção para que ele aponte para o nó que sucede o nó a ser removido. Em seguida, removemos o nó.





Listas Encadeadas: Remoção

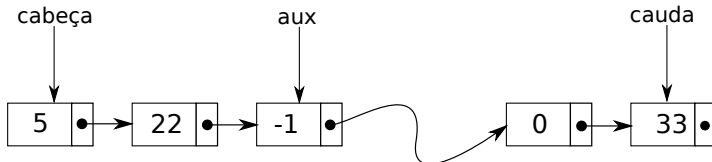
Para remover um elemento arbitrário, devemos percorrer a lista até o nó anterior a posição de remoção para que ele aponte para o nó que sucede o nó a ser removido. Em seguida, removemos o nó.





Listas Encadeadas: Remoção

Para remover um elemento arbitrário, devemos percorrer a lista até o nó anterior a posição de remoção para que ele aponte para o nó que sucede o nó a ser removido. Em seguida, removemos o nó.



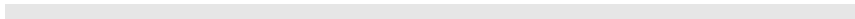


Listas Encadeadas: Remoção





Listas Encadeadas: Remoção





Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- Remoção
- **Acesso**
- Limpeza
- Análise



Listas Encadeadas: Acesso

- Acesso na cabeça ou na cauda é fácil. Já temos ponteiros para estas posições.
- Para acessar uma posição arbitrária, começamos da cabeça e iteramos na lista até posicionarmos o ponteiro na posição em que queremos acessar.
- Diferentemente de vetores, listas encadeadas não possuem acesso direto (aleatório).



Listas Encadeadas: Acesso

Acesso a cabeça da lista.





Listas Encadeadas: Acesso

Acesso a cauda.





Listas Encadeadas: Acesso

Acesso a uma posição arbitrária.





Listas Encadeadas: Acesso

Acesso a uma posição arbitrária.





Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- Remoção
- Acesso
- **Limpeza**
- Análise



Listas Encadeadas: Limpeza

- Para deletar a lista da memória, basta iterar sobre ela e apagar os nós.
- Só devemos ter cuidado de não perder a referência para o próximo nó.
- Uma estratégia é sempre apagar a cabeça da lista enquanto ela não é vazia.



Listas Encadeadas: Limpeza





Sumário

2 Listas Encadeadas

- Definição
- Inicialização
- Funções auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Encadeadas

Operação	Complexidade
Inserção na cabeça	$\Theta(1)$
Inserção na cauda	$\Theta(1)$
Inserção em posição arbitrária	$\Theta(n)$
Remoção da cabeça	$\Theta(1)$
Remoção da cauda	$\Theta(n)$
Remoção de uma posição arbitrária	$\Theta(n)$
Acesso à cabeça	$\Theta(1)$
Acesso à cauda	$\Theta(1)$
Acesso à posição arbitrária	$\Theta(n)$



Sumário

3 Listas Duplamente Encadeadas



Listas Duplamente Encadeadas

- Listas duplamente encadeadas se assemelham muito às listas encadeadas com a diferença que cada elemento possui uma referência para o elemento anterior.
- Apesar de utilizar mais espaço para representação, pode-se caminhar no sentido contrário.
- As operações em Listas Duplamente encadeada são **similares** às das Listas Encadeadas, com atenção para atualizar o ponteiro do elemento anterior.



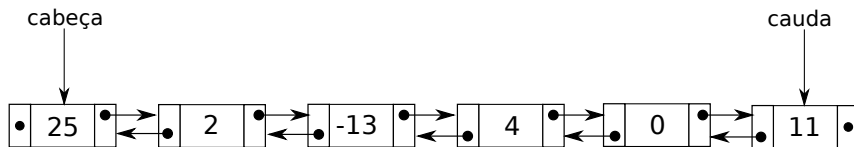
Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Duplamente Encadeadas





Listas Duplamente Encadeadas: Definição

```
7  /**A nossa dlista encadeada consiste de vários nós,  
8  que possuem o tipo dlist_node_t */  
9  typedef struct dlist_node_t {  
10     int data; /*Dado*/  
11     struct dlist_node_t *next; /*ponteiro para o próximo elemento*/  
12     struct dlist_node_t *prev; /*Ponteiro para o elemento anterior*/  
13 } dlist_node_t;
```



Listas Duplamente Encadeadas: Definição

```
15 typedef struct dlist_t {  
16     dlist_node_t *head; /*Cabeça da dlista*/  
17     dlist_node_t *tail; /*Cauda da dlista*/  
18     size_t size;        /*tamanho da dlista*/  
19 } dlist_t;
```



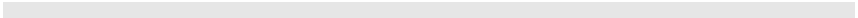
Sumário

3 Listas Duplamente Encadeadas

- Definição
- **Inicialização**
- Funções Auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Duplamente Encadeadas: Inicialização





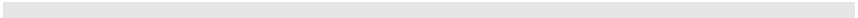
Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- **Funções Auxiliares**
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise

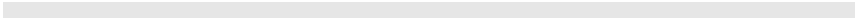


Listas Duplamente Encadeadas: Funções Auxiliares





Listas Duplamente Encadeadas: Funções Auxiliares





Listas Duplamente Encadeadas: Funções Auxiliares





Listas Duplamente Encadeadas: Funções Auxiliares





Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- **Inserção**
- Remoção
- Acesso
- Limpeza
- Análise



Listas Duplamente Encadeadas: Inserção

Inserção na Cabeça e Cauda

- Igual às versões das listas encadeadas.
- Só precisamos de cuidado para atualizar os ponteiros que ligam ao próximo ou ao anterior.

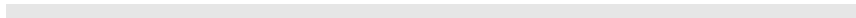


Listas Duplamente Encadeadas: Inserção na Cabeça





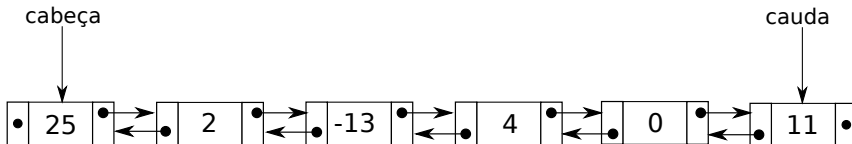
Listas Duplamente Encadeadas: Inserção na Cauda





Listas Duplamente Encadeadas: Inserção

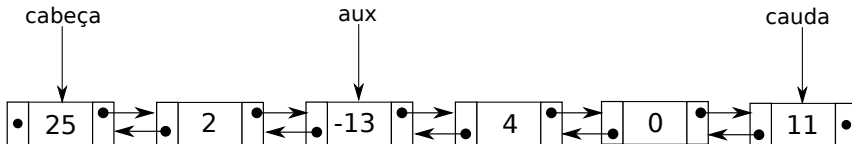
Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção

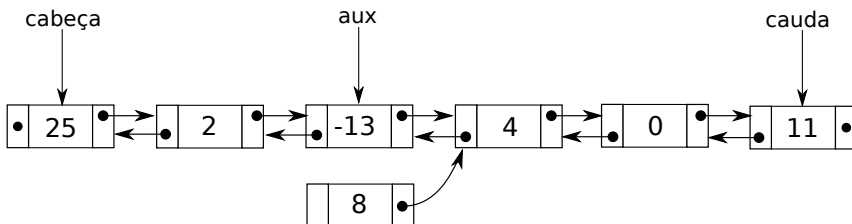
Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção

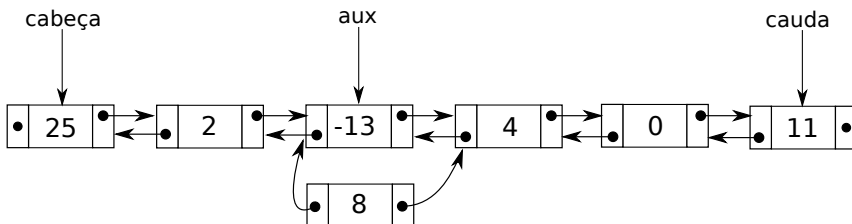
Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção

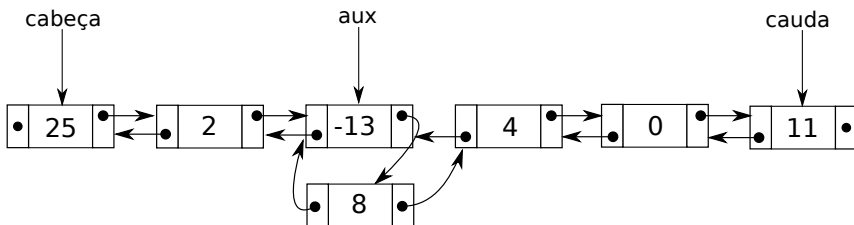
Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção

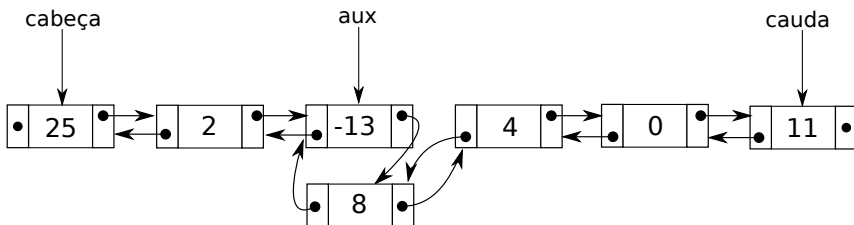
Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção

Percorremos a lista até chegar na posição que antecede a inserção para conseguirmos encaixar o novo nó entre dois nós existentes.





Listas Duplamente Encadeadas: Inserção





Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- Inserção
- **Remoção**
- Acesso
- Limpeza
- Análise

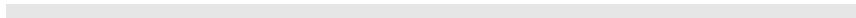


Listas Duplamente Encadeadas: Remoção na Cabeça

- Igual a versão das listas encadeadas.
- Só precisamos de cuidado para atualizar os ponteiros adicionais.



Listas Duplamente Encadeadas: Remoção na Cabeça





Listas Duplamente Encadeadas: Remoção na Cauda

- Na versão da lista encadeada simples, precisávamos percorrer a lista toda até o penúltimo elemento.
- Como em listas duplamente encadeadas conseguimos acessar o penúltimo elemento ao começar do último e utilizar o ponteiro para o anterior, o penúltimo elemento é obtido em tempo constante!
- $\Theta(n) \Rightarrow \Theta(1)$.
- O restante da remoção é igual ao da lista encadeada simples, com cuidado de atualizar os ponteiros adicionais.



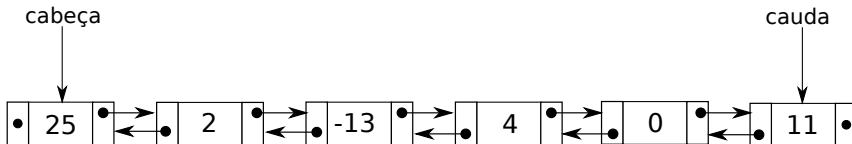
Listas Duplamente Encadeadas: Remoção na Cauda





Listas Duplamente Encadeadas: Remoção

Para remoção em uma posição arbitrária, percorremos a lista até chegar na posição que queremos remover.





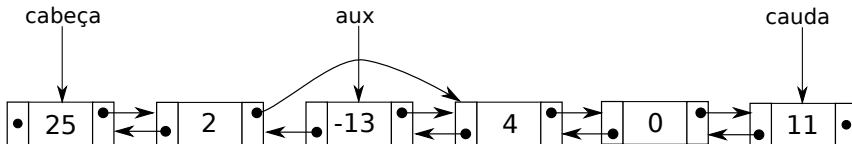
Listas Duplamente Encadeadas: Remoção

Para remoção em uma posição arbitrária, percorremos a lista até chegar na posição que queremos remover.



Listas Duplamente Encadeadas: Remoção

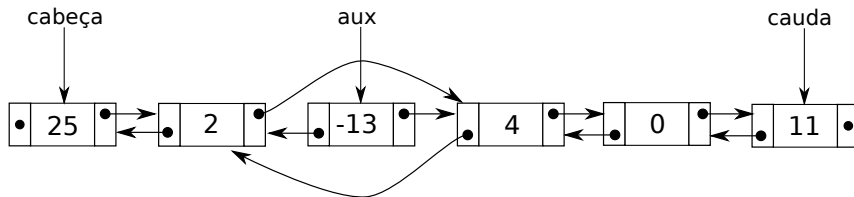
Para remoção em uma posição arbitrária, percorremos a lista até chegar na posição que queremos remover.





Listas Duplamente Encadeadas: Remoção

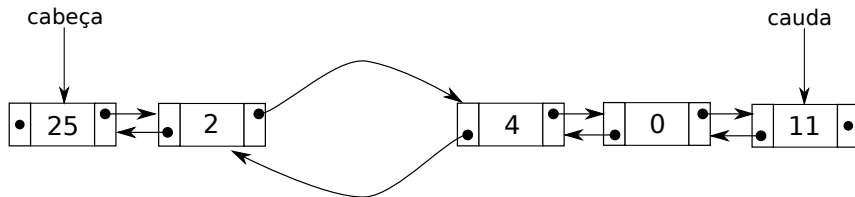
Para remoção em uma posição arbitrária, percorremos a lista até chegar na posição que queremos remover.





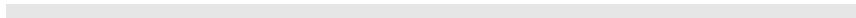
Listas Duplamente Encadeadas: Remoção

Para remoção em uma posição arbitrária, percorremos a lista até chegar na posição que queremos remover.





Listas Duplamente Encadeadas: Inserção





Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- Inserção
- Remoção
- **Acesso**
- Limpeza
- Análise

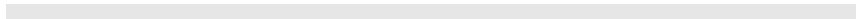


Listas Duplamente Encadeadas: Acesso

- O Acesso em listas duplamente encadeadas é análogo ao das listas encadeadas simples.
- Durante o acesso à uma posição arbitrária, podemos começar a busca pela cabeça ou pela cauda, a escolha dependerá de qual estará mais próxima da posição em que se deseja acesso.

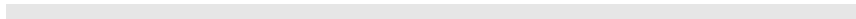


Listas Duplamente Encadeadas: Acesso





Listas Duplamente Encadeadas: Acesso





Listas Duplamente Encadeadas: Acesso





Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- Inserção
- Remoção
- Acesso
- **Limpeza**
- Análise



Listas Duplamente Encadeadas: Limpeza

- Funciona de forma análoga ao das listas encadeadas simples.



Listas Duplamente Encadeadas: Limpeza





Sumário

3 Listas Duplamente Encadeadas

- Definição
- Inicialização
- Funções Auxiliares
- Inserção
- Remoção
- Acesso
- Limpeza
- Análise



Listas Duplamente Encadeadas

Complexidade das Operações

Operação	Complexidade
Inserção na cabeça	$\Theta(1)$
Inserção na cauda	$\Theta(1)$
Inserção em posição arbitrária	$\Theta(n)$
Remoção da cabeça	$\Theta(1)$
Remoção da cauda	$\Theta(1)$
Remoção de uma posição arbitrária	$\Theta(n)$
Acesso à cabeça	$\Theta(1)$
Acesso à cauda	$\Theta(1)$
Acesso à posição arbitrária	$\Theta(n)$

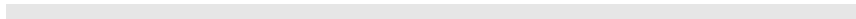


Sumário

4 Exemplos



Exemplo da Utilização da Biblioteca





Exemplo da Utilização da Biblioteca

