

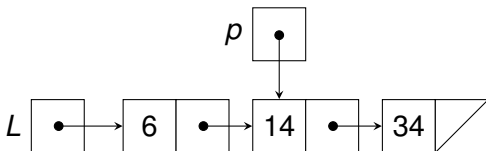
Estrutura de Dados I

Luciana Lee

Tópicos da Aula

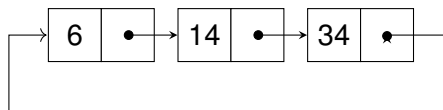
- 1 Lista Circular Dinâmica
- 2 Lista Circular Estática

Lista Circular Dinâmica



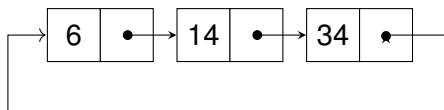
- Dado um ponteiro p para um nó numa lista linear, não podemos atingir nenhum dos nós que antecedem p .
- Suponha que seja feita uma pequena mudança na estrutura de uma lista simplesmente encadeada, de modo que o campo *prox* no último nó contenha um ponteiro de volta para o primeiro nó, em vez de um ponteiro nulo.

Lista Circular Dinâmica



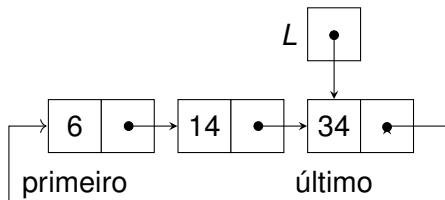
- Esse tipo de lista é chamado **lista circular**.

Lista Circular Dinâmica



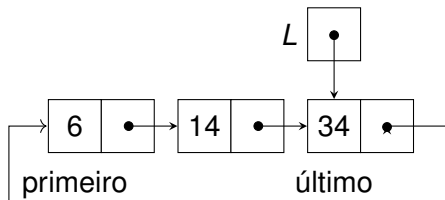
- Esse tipo de lista é chamado **lista circular**.
- Observe que uma lista circular não tem um “primeiro” ou um “último” nó natural.

Lista Circular Dinâmica



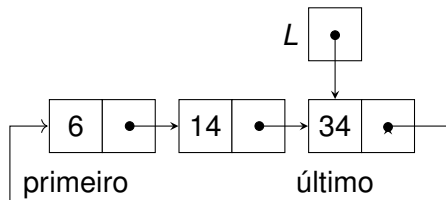
- Esse tipo de lista é chamado **lista circular**.
- Observe que uma lista circular não tem um “primeiro” ou um “último” nó natural.
- Vamos adotar como convenção que o ponteiro da lista guarde o endereço do último elemento.

Lista Circular Dinâmica



- Esse tipo de lista é chamado **lista circular**.
- Observe que uma lista circular não tem um “primeiro” ou um “último” nó natural.
- Vamos adotar como convenção que o ponteiro da lista guarde o endereço do último elemento.
- Qual é a vantagem dessa convenção?

Lista Circular Dinâmica



- Notem que o último elemento é acessado por L ;
- E o primeiro elemento é acessado por $L \rightarrow prox.$
- Quando a lista está vazia: $L = NULL$;

Operações Básicas

- Inserir no início da lista
- Inserir no final da lista
- Excluir o primeiro da lista
- Excluir o último da lista
- Imprimir lista

Lista Circular Dinâmica: Vantagens

- Podemos varrer a lista completamente, a partir de qualquer ponto e voltar ao início, a partir de qualquer ponto da lista
- Algumas operações (como a concatenação) são mais eficazes.

Aplicação: O Problema de Josephus I

- Um grupo de soldados está cercado por um exército inimigo devastador.
- Não há esperanças de vitória sem a chegada de reforços, mas existe somente um cavalo disponível para escapar.
- Os soldados entram em um acordo para determinar qual deles deverá escapar e trazer ajuda.
- Eles formam um círculo e um número n é sorteado num chapéu. Um de seus nomes é sorteado também.
- Começando pelo soldado cujo nome foi sorteado, eles começam a contar ao longo do círculo em sentido horário.
- Quando a contagem alcança n , esse soldado é retirado do círculo, e a contagem reinicia com o soldado seguinte.

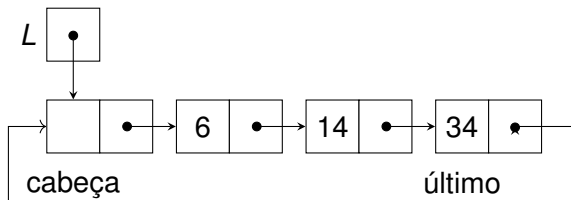
Aplicação: O Problema de Josephus II

- O processo continua de maneira que, toda vez que n é alcançado, outro soldado sai do círculo.
- Todo soldado retirado do círculo não entra mais na contagem.
- O último soldado que restar deverá montar no cavalo e escapar.
- Considerando um número n , a ordenação dos soldados no círculo e o soldado a partir do qual começa a contagem, o problema é determinar a seqüência na qual os soldados são eliminados do círculo e o soldado que escapará.

Aplicação: O Problema de Josephus

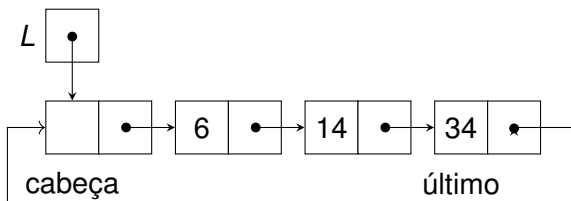
- A entrada para o programa é o número n e uma lista de nomes, que será o seqüenciamento do círculo em sentido horário, começando pelo soldado a partir do qual a contagem deve ser iniciada.
- A última linha de entrada contém a string “END” indicando o final da entrada.
- O programa deve imprimir os nomes na seqüência em que são eliminados e o nome do soldado que escapará.

Lista Circular Dinâmica com Nó Cabeça



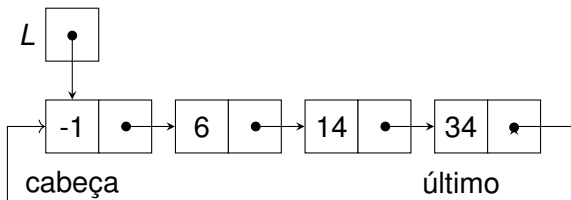
- Uma lista circular pode ser implementada com um nó cabeça!

Lista Circular Dinâmica com Nó Cabeça



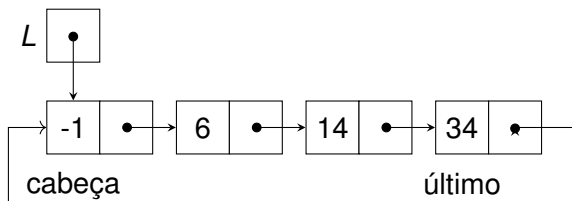
- Uma lista circular pode ser implementada com um nó cabeça!
- Podemos colocar o nó cabeça como primeiro nó da lista circular.

Lista Circular Dinâmica com Nó Cabeça



- Uma lista circular pode ser implementada com um nó cabeça!
- Podemos colocar o nó cabeça como primeiro nó da lista circular.
- Esse cabeçalho de lista pode ser reconhecido por um valor especial em seu campo *info* que não será um conteúdo válido de um nó de lista no contexto do problema, ou ele poderá conter um sinal marcando-o como cabeçalho.

Lista Circular Dinâmica com Nó Cabeça



- Uma lista circular pode ser implementada com um nó cabeça!
- Podemos colocar o nó cabeça como primeiro nó da lista circular.
- Esse cabeçalho de lista pode ser reconhecido por um valor especial em seu campo *info* que não será um conteúdo válido de um nó de lista no contexto do problema, ou ele poderá conter um sinal marcando-o como cabeçalho.
- Poderemos percorrer a lista usando um único ponteiro, com o percurso interrompido quando o nó cabeça for alcançado.

Lista Circular Dinâmica com Nó Cabeça

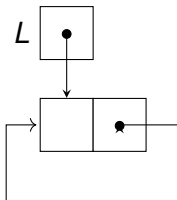
- Se, ao invés de utilizarmos o próprio ponteiro da lista para percorrê-lo, utilizarmos um ponteiro auxiliar para tal tarefa, então não precisamos armazenar um código especial no nó cabeça.

Lista Circular Dinâmica com Nó Cabeça

- Se, ao invés de utilizarmos o próprio ponteiro da lista para percorrê-lo, utilizarmos um ponteiro auxiliar para tal tarefa, então não precisamos armazenar um código especial no nó cabeça.
- O final do percurso seria marcado pela igualdade entre o ponteiro auxiliar e o ponteiro do nó cabeça.

Lista Circular Dinâmica com Nó Cabeça

- Se, ao invés de utilizarmos o próprio ponteiro da lista para percorrê-lo, utilizarmos um ponteiro auxiliar para tal tarefa, então não precisamos armazenar um código especial no nó cabeça.
- O final do percurso seria marcado pela igualdade entre o ponteiro auxiliar e o ponteiro do nó cabeça.
- Como é a lista vazia?

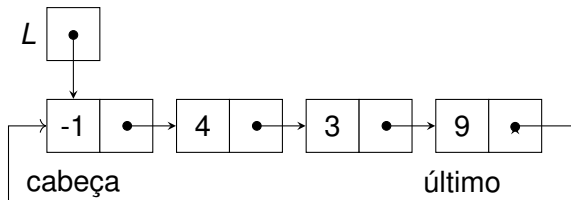


Aplicação: Soma de Inteiros Positivos Longos

- Em Pascal, o tipo integer pode ser de -32.768 a 32.767 ou mesmo ter 32 bits variando de 2147483648 a 2147483647 .
- **Long int** tem 64 bits, podendo variar de $9223372036854775808 \dots 9223372036854775807$
- Variação parecida ocorre com C
- Para somar números maiores: Podemos utilizar listas!

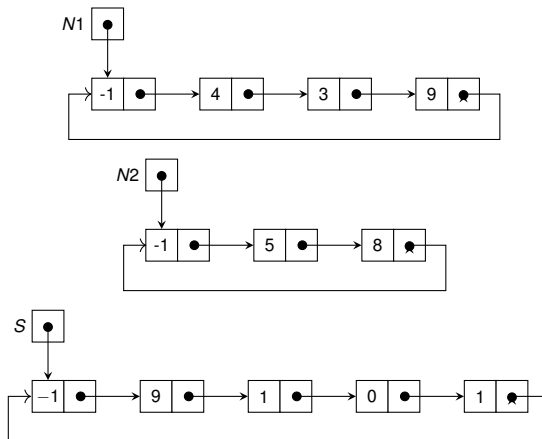
Aplicação: Soma de Inteiros Positivos Longos

- Para somar dois inteiros longos desse tipo, seus dígitos são atravessados da direita para a esquerda, e os dígitos correspondentes e um possível transporte da soma dos dígitos anteriores são acrescidos.
- Isso sugere a representação de inteiros longos, armazenando seus dígitos da direita para a esquerda numa lista, de modo que o primeiro nó na lista contenha o dígito menos significativo (o da extrema direita) e o último nó contenha o dígito mais significativo (o da extrema esquerda).
- Exemplo: O número 934 é representado pela lista:



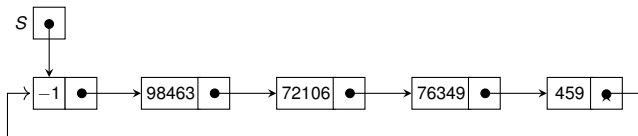
Aplicação: Soma de Inteiros Positivos Longos

- A soma dos números $N1 = 934$ e $N2 = 85$ resulta em:



Aplicação: Soma de Inteiros Positivos Longos

- Para economizar espaço, manteremos cinco dígitos em cada nó. Ou seja, o valor máximo representado em um nó é 99999.
- Assim, o número 459763497210698463 é representado pela lista:



Aplicação: Soma de Inteiros Positivos Longos

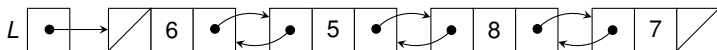
- Ambas as listas são atravessadas paralelamente, e cinco dígitos são somados por vez.
- Se a soma de dois números de cinco dígitos for x , os cinco dígitos de menor ordem de x poderão ser extraídos usando a expressão $x \% 100000$, que resulta no resto de x na divisão por 100000.
- O transporte pode ser calculado pela divisão de inteiros $x / 100000$.
- Quando o final de uma das listas for alcançado, o transporte será passado para os dígitos restantes da outra lista.

Listas Duplamente Encadeadas Circulares

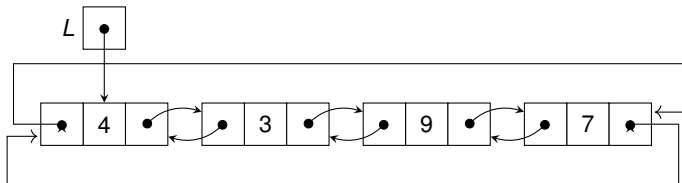
- Embora uma lista circular simplesmente encadeada tenha vantagens sobre uma lista simplesmente encadeada não circular, ela ainda apresenta várias deficiências.
- Não se pode atravessar uma lista desse tipo no sentido contrário nem um nó pode ser eliminado de uma lista circularmente ligada, em função de apenas um ponteiro para esse nó.
- Nos casos em que tais recursos são necessários, a estrutura de dados adequada é uma lista duplamente encadeada.

Listas Duplamente Encadeadas Circulares

- Como vimos, a estrutura de uma lista duplamente encadeada não circular é:

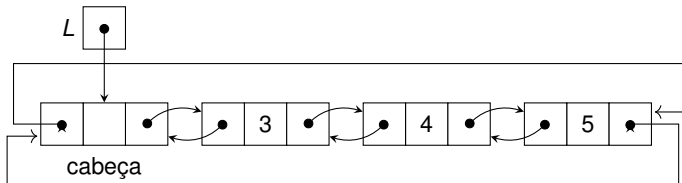


- A estrutura de uma lista duplamente encadeada circular é:



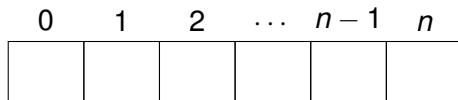
Listas Duplamente Encadeadas Circulares

- A estrutura de uma lista duplamente encadeada circular com nó cabeça é:



Lista Circular Estática

- Para implementar uma estrutura de lista circular estática utilizaremos um vetor.
- Como já vimos, o vetor possui pouca flexibilidade quanto ao seu tamanho, mas permite um acesso direto aos elementos (utilizando os índices).
- Imagine um vetor com capacidade para armazenar n elementos.

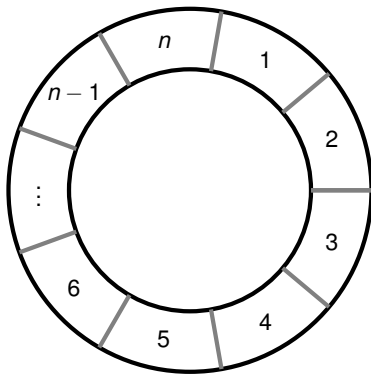


Lista Circular Estática

- Vamos analisar como fazemos as operações na estrutura descrita:
 - ▶ Inserir no início
 - ▶ Inserir no final
 - ▶ Remover no início
 - ▶ Remover no final

Lista Circular Estática

- Agora, vamos imaginar que nosso vetor pode ser trabalhado de “forma circular”.
- Imagine que o vetor possui o formato de uma roda, como mostra a imagem.



Lista Circular Estática

- Vamos trabalhar com dois índices: `inicio` e `fim`.
- Também precisaremos do número de elementos que ocupam a lista.
- Vamos analisar os casos para:
 - ▶ Inserir no início
 - ▶ Inserir no final
 - ▶ Remover no início
 - ▶ Remover no final

Tarefas

- 1 Implemente um programa que resolva o problema de Josephus. Considere a descrição do problema apresentado em aula.
- 2 Implemente um programa que soma valores inteiros positivos muito grandes (exemplo descrito em aula). Considere que cada elemento da lista armazena valores com até 5 dígitos. Seu programa vai ler dois números inteiros e imprimir o resultado da soma.
- 3 Implemente uma lista circular dinâmica com lista duplamente encadeada. As operações de inclusão, exclusão e impressão analisadas em aula devem ser implementadas.
- 4 Implemente uma lista circular estática com as operações analisadas em aula.

Dúvidas?