

Lição 7



Listas

Objetivos

Ao final desta lição, o estudante será capaz de:

- Explicar as definições e conceitos básicos de listas
- Usar as diferentes representações de lista: seqüencial e encadeada
- Diferenciar lista encadeada simples, lista encadeada dupla, lista circular e lista com *header nodes*
- Explicar como as listas são aplicadas na aritmética polinomial
- Discutir as estruturas de dado usadas na alocação dinâmica de memória usando métodos *sequential-fit* e métodos *buddy-system*



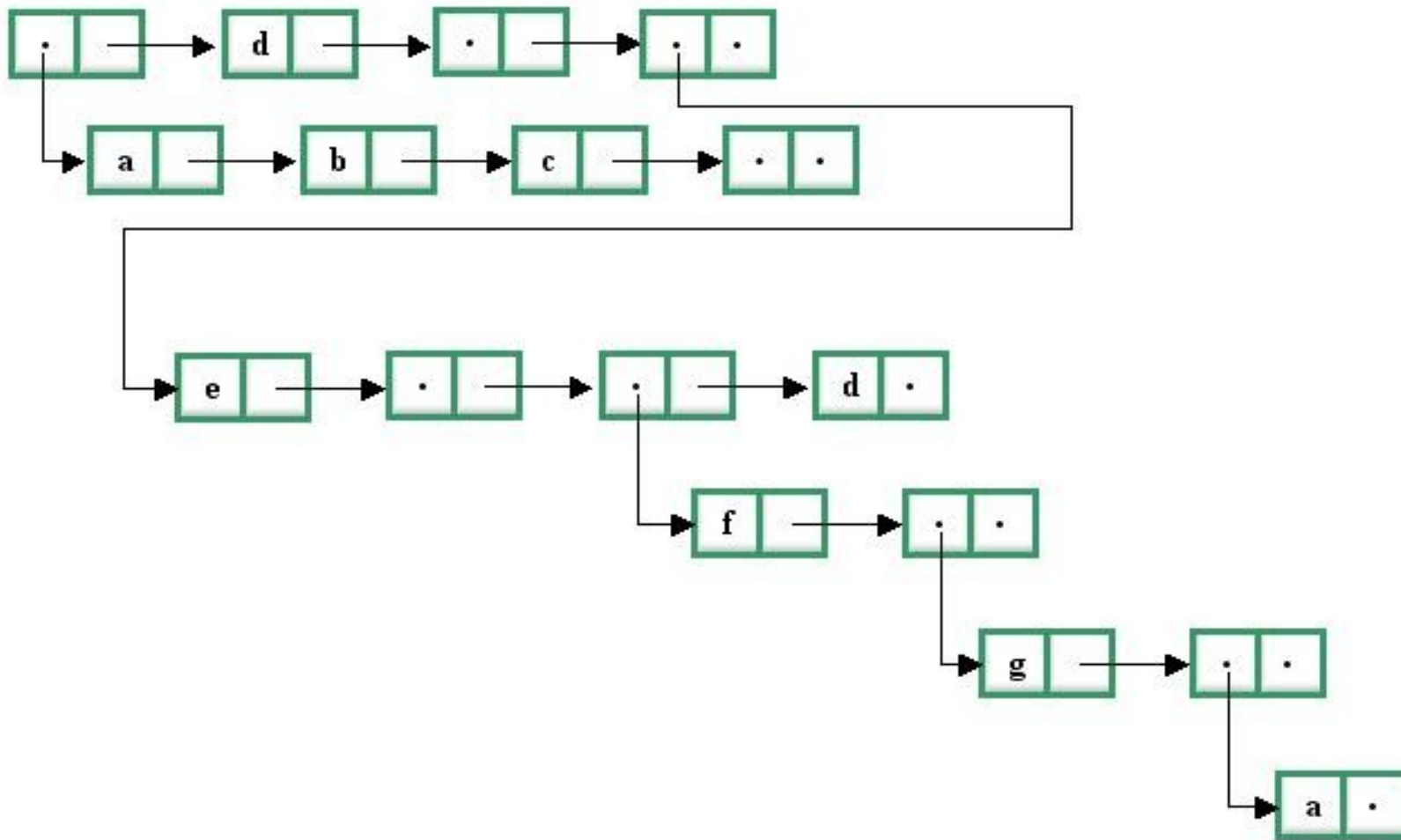
Definição e Conceitos Relacionados

- Conjunto finito de zero ou mais elementos
- Elementos podem ser *atômicos* ou *listados*
 - *Lista Linear*
 - *Lista Generalizada*



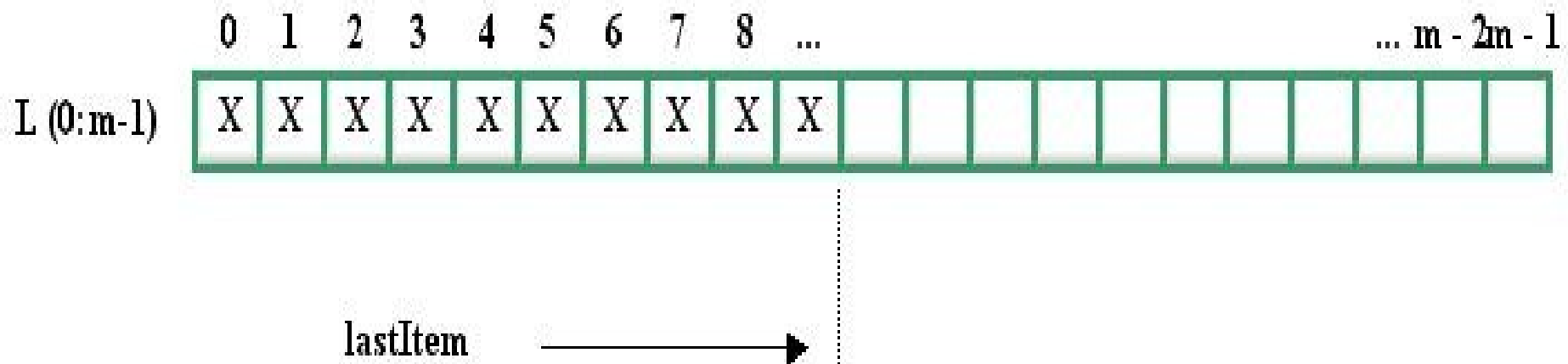
Definição e Conceitos

Relacionados: Lista Generalizada



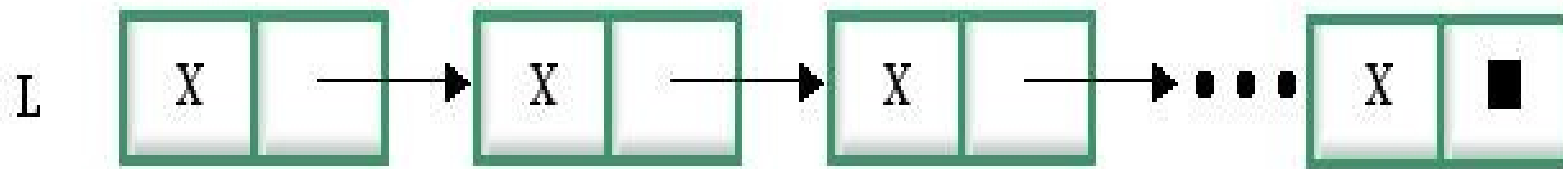
Representações de Listas

- Apropriada para listas naturalmente estáticas



Representações de Listas

- Representação “*linkada*” de lista Linear *Singly-Linked*
- Adequada para listas com tamanho desconhecido



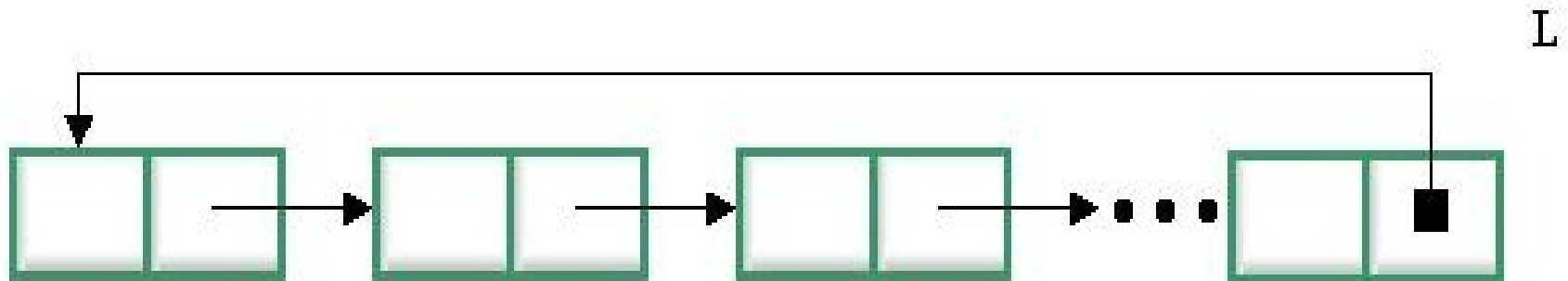
Representações de Listas

<i>Operação</i>	<i>Representação Seqüencial</i>	<i>Representação Linkada</i>
Determinando se uma lista é vazia	$O(1)$	$O(1)$
Encontrando o comprimento	$O(1)$	$O(n)$
Acessando o $n^{\text{ésimo}}$ elemento	$O(1)$	$O(n)$
Atualizando o $n^{\text{ésimo}}$ elemento	$O(1)$	$O(n)$
Deletando o $n^{\text{ésimo}}$ elemento	$O(n)$	$O(n)$
Inserindo um novo elemento	$O(n)$	$O(1)$



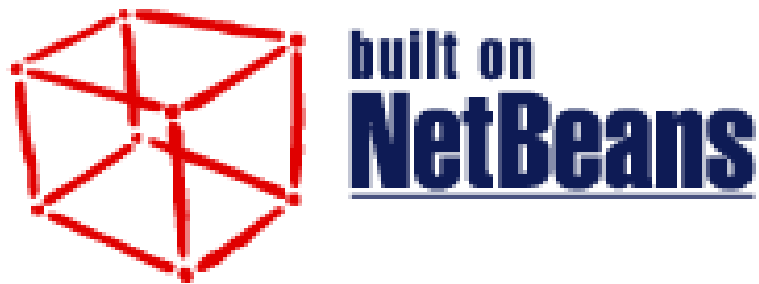
Representações de Listas: Lista Circular *Singly-Linked*

- Formada pela configuração do *link* no último *node* apontando para o primeiro *node*



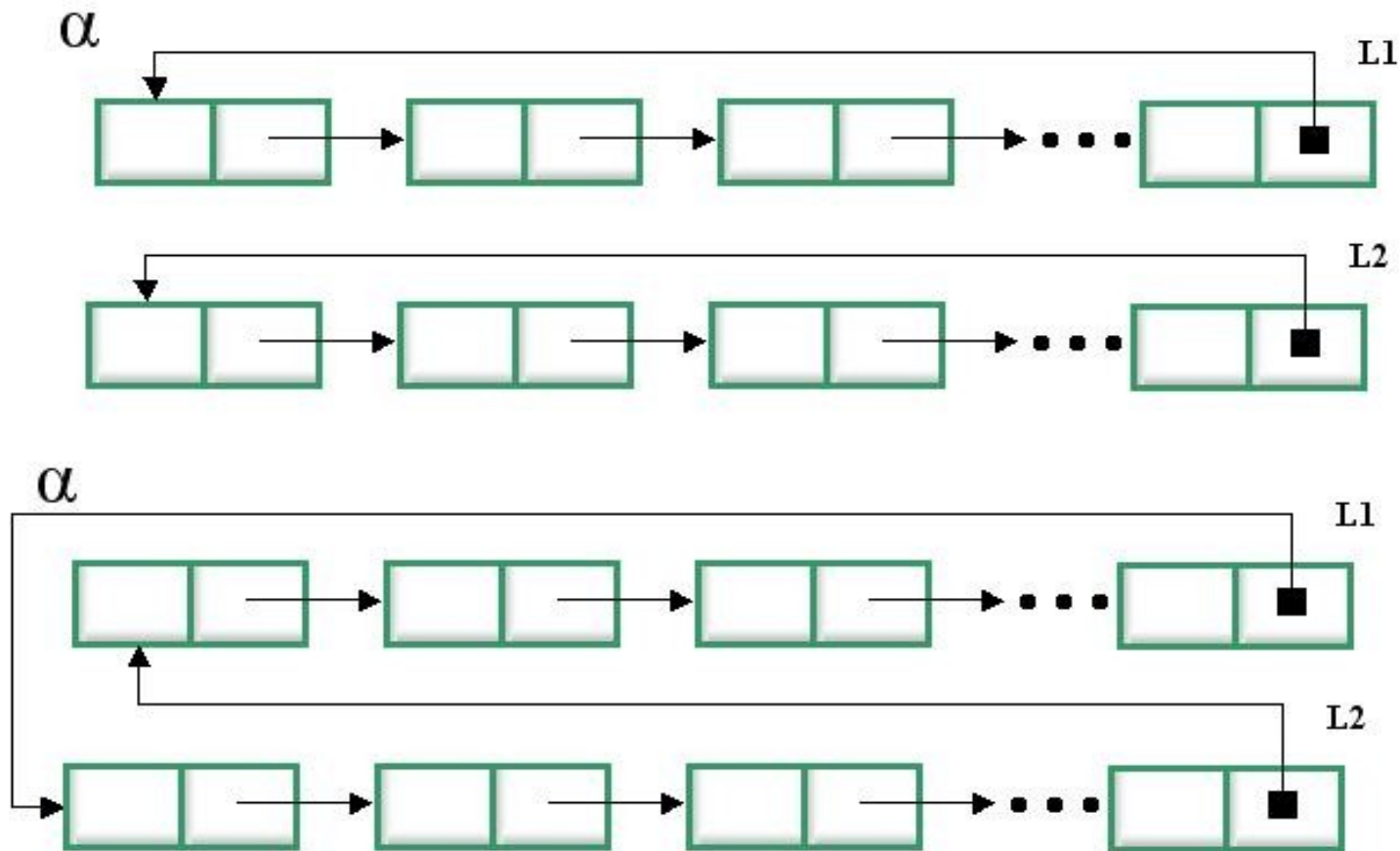
Representações de Listas: Lista Circular *Singly-Linked*

- Passaremos agora para o NetBeans



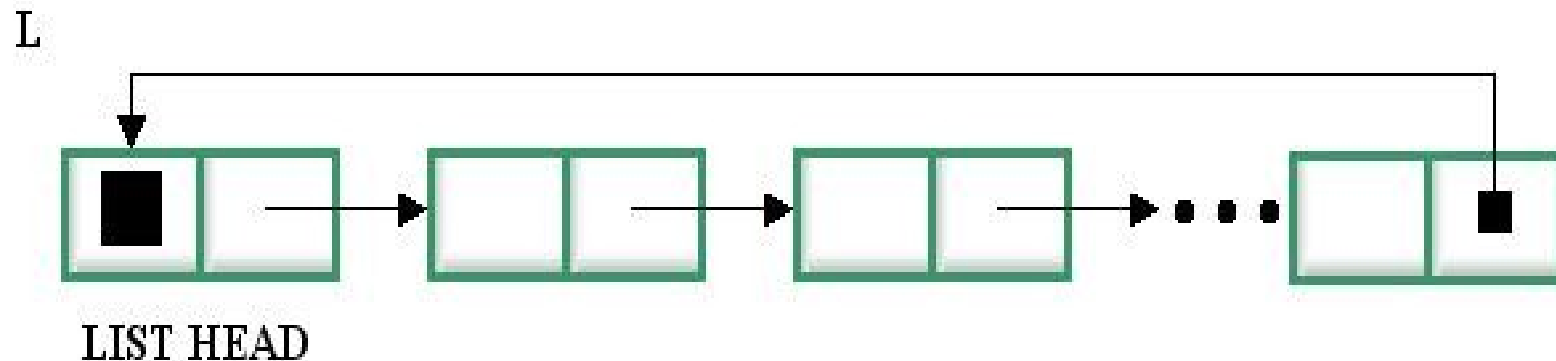
Representações de Listas: Lista Circular *Singly-Linked*

- Operação de Concatenação:



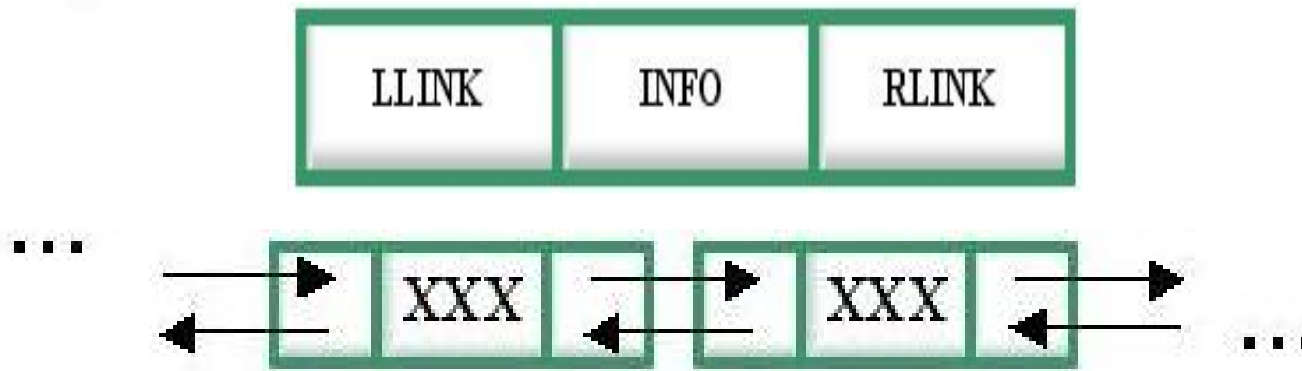
Representações de Listas: Lista Circular *Singly-Linked*

- Lista *Singly-Linked* com *Header Node*
- *Header node* é também conhecido como ***list head***
- *List head* serve como um sentinela em uma lista circular para indicar preenchimento transversal da lista



Representações de Listas: Lista *Double-Linked*

- *Nodes* têm ponteiros para ambos os vizinhos da lista, esquerda e direita
- Pode ser constituída da seguinte forma:
 - Lista linear *Doubly-linked*
 - Lista circular *Doubly-linked*
 - Lista circular com *list head Doubly-linked*



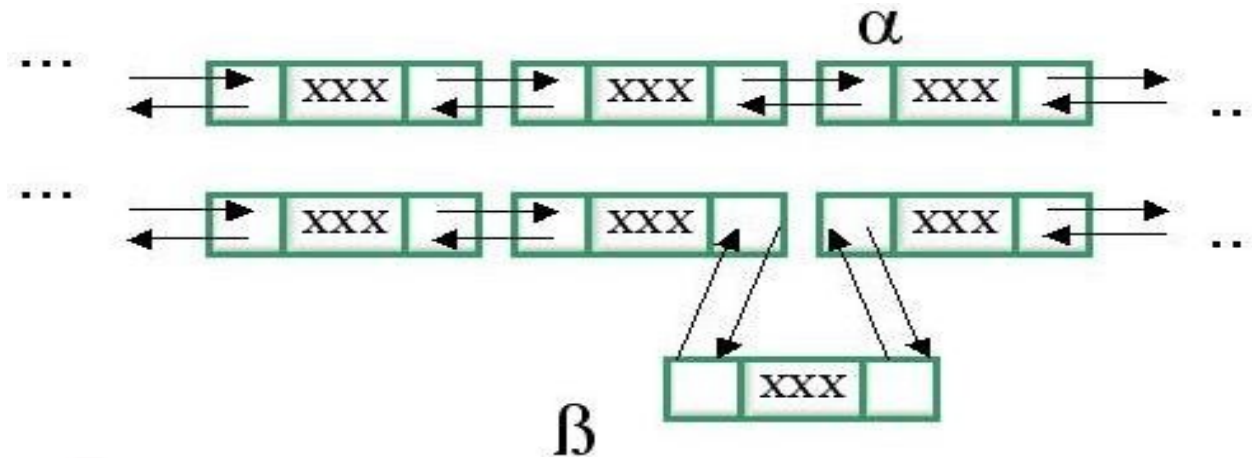
Representações de Listas: Lista *Double-Linked*

- Propriedades:
 - $LLINK(L) = RLINK(L) = L$ significa que a lista L está vazia
 - Pode inserir um novo *node*, digamos o *node* β , à esquerda (ou direita) de qualquer *node*, digamos o *node* α , no tempo $O(1)$, conhecendo apenas α
 - Pode eliminar qualquer *node*, diz-se *node* α , em L no tempo $O(1)$, conhecendo apenas o endereço α

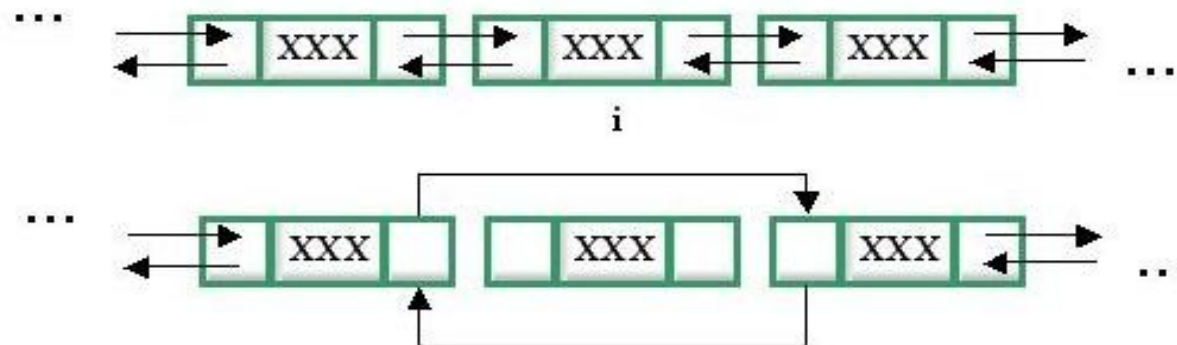


Representações de Listas: Lista *Double-Linked*

- Inserção



- Deleção



Aplicação: Aritmética Polinomial

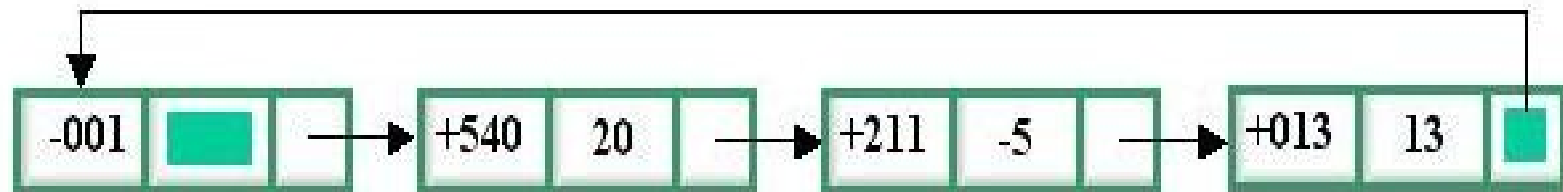
- Características para negociar
- Lista circular *Singly-linked* com *list head* pode ser usada

EXPO				COEF	LINK
S	e_x	e_y	e_z		

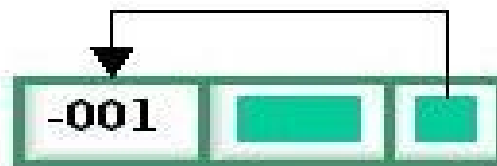
Aplicação: Aritmética Polinomial

- Exemplo: $P(x,y,z) = 20x^5y^4 - 5x^2yz + 13yz^3$

P



- Representação do zero polinomial:

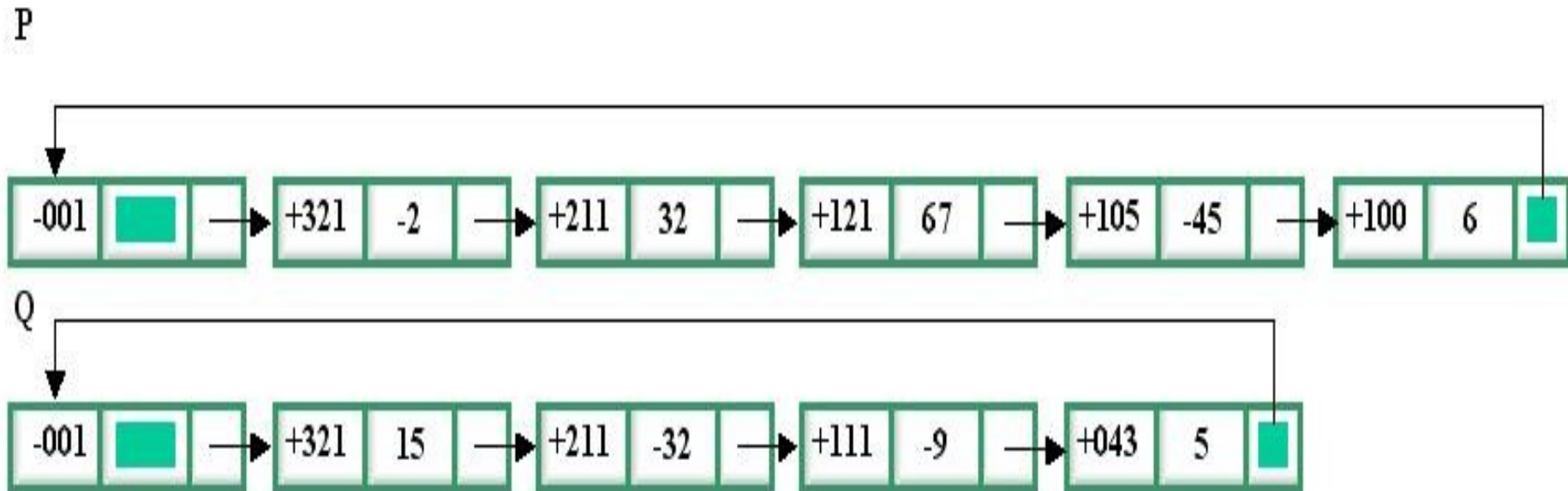


$$P(x,y,z) = 0$$

Estruturas de Dados

Aplicação: Aritmética Polinomial

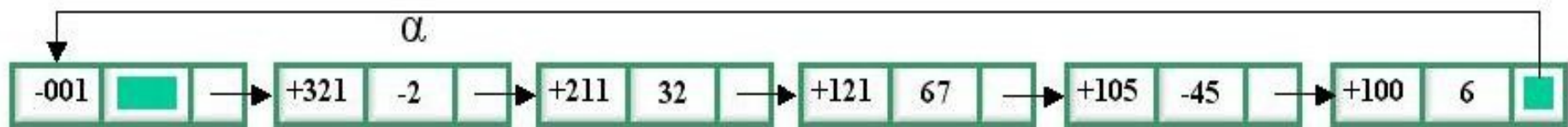
- Adição Polinomial ($P + Q$)



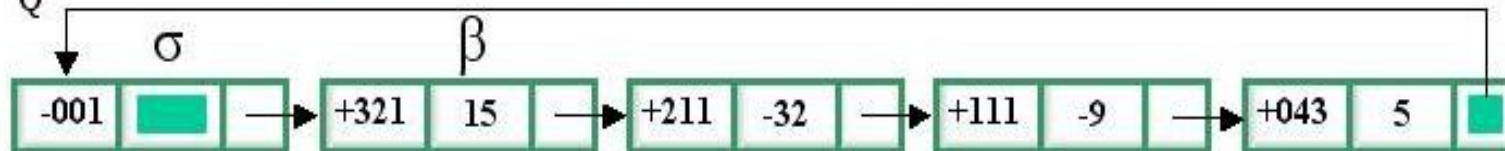
Aplicação: Aritmética Polinomial

- Adição Polinomial ($P + Q$)

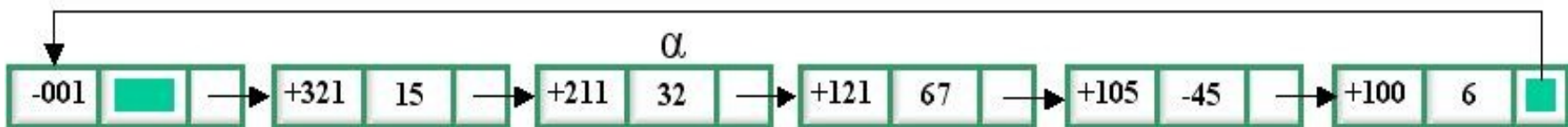
P



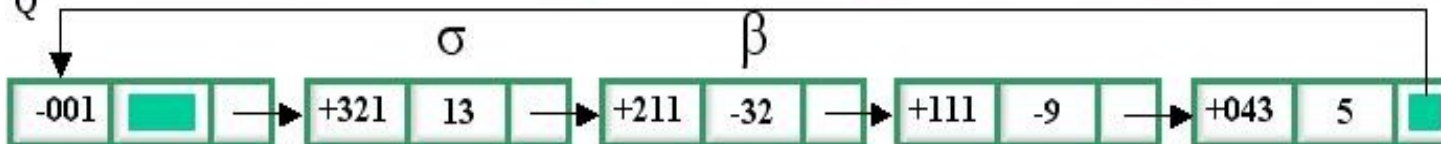
Q



P

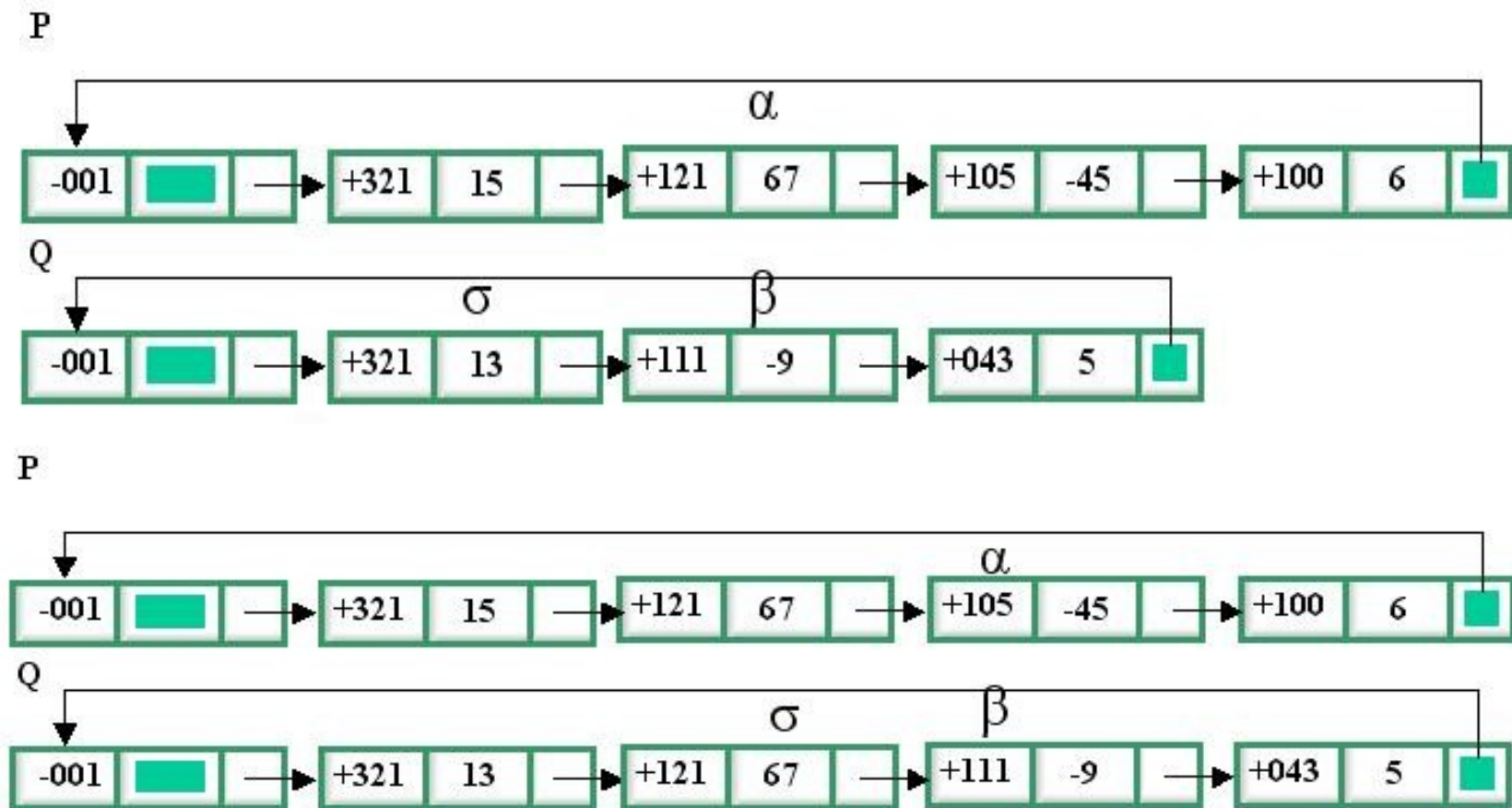


Q



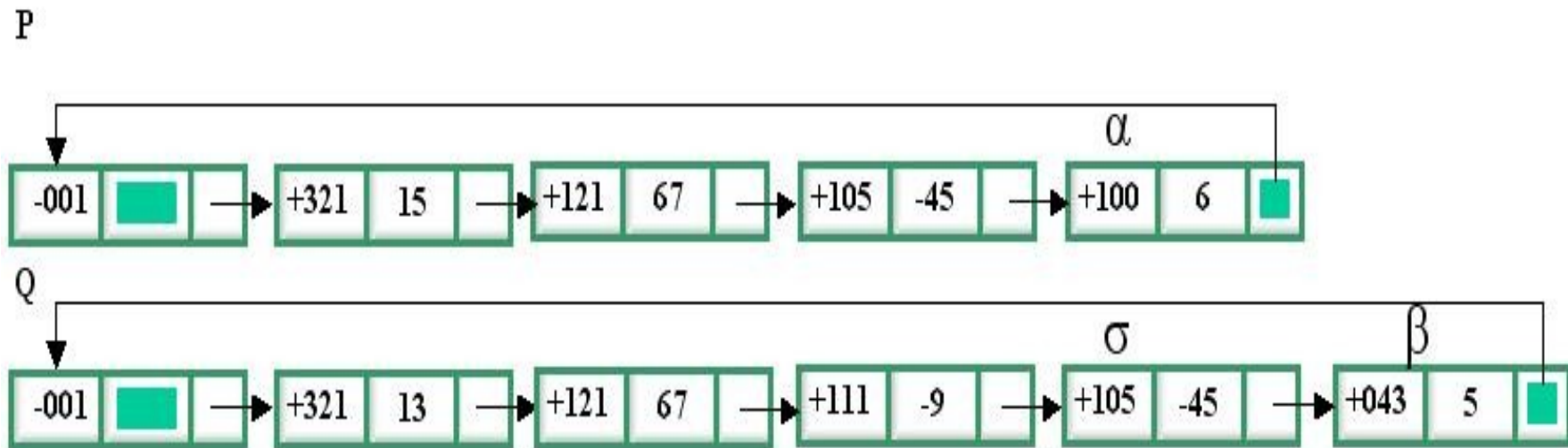
Aplicação: Aritmética Polinomial

- Adição Polinomial ($P + Q$)



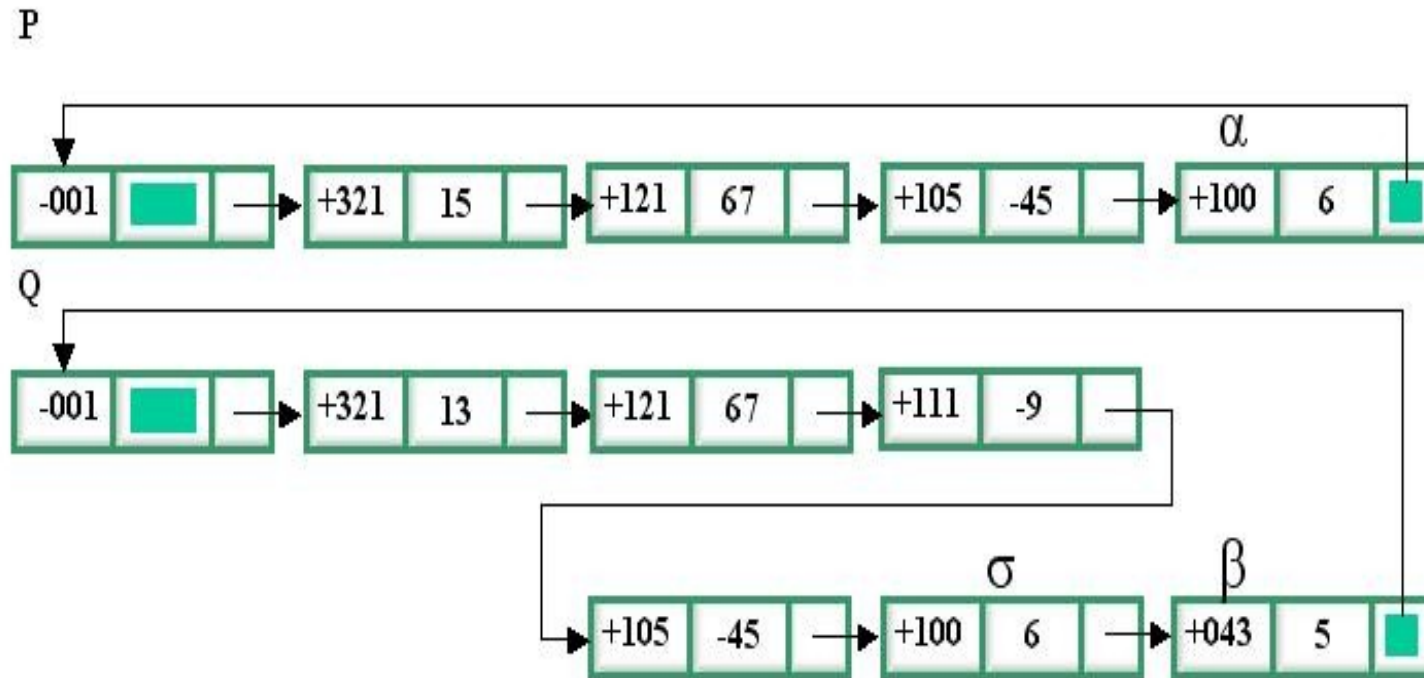
Aplicação: Aritmética Polinomial

- Adição Polinomial ($P + Q$)



Aplicação: Aritmética Polinomial

- Adição Polinomial ($P + Q$)



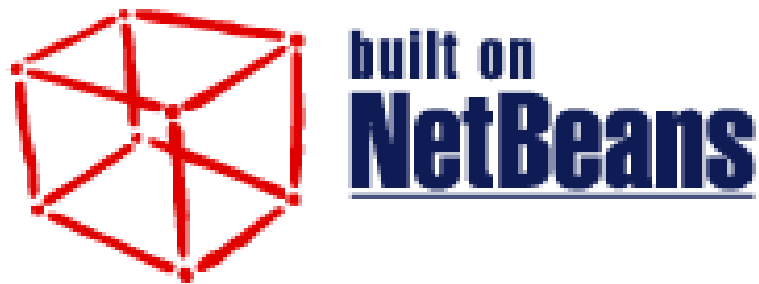
Aplicação: Aritmética Polinomial

- Subtração Polinomial
 - $Q = P - Q = P + (-Q)$
 - Percorre Q e nega os coeficientes adicionados ao polinômio
- Multiplicação Polinomial
 - Ao se multiplicar dois polinômios P e Q, um zero polinomial R é necessário para receber o produto, i.e. $R = R + P * Q$
 - Todo termo em P é multiplicado com todo termo em Q



Aplicação: Aritmética Polinomial

- Passaremos agora para o NetBeans



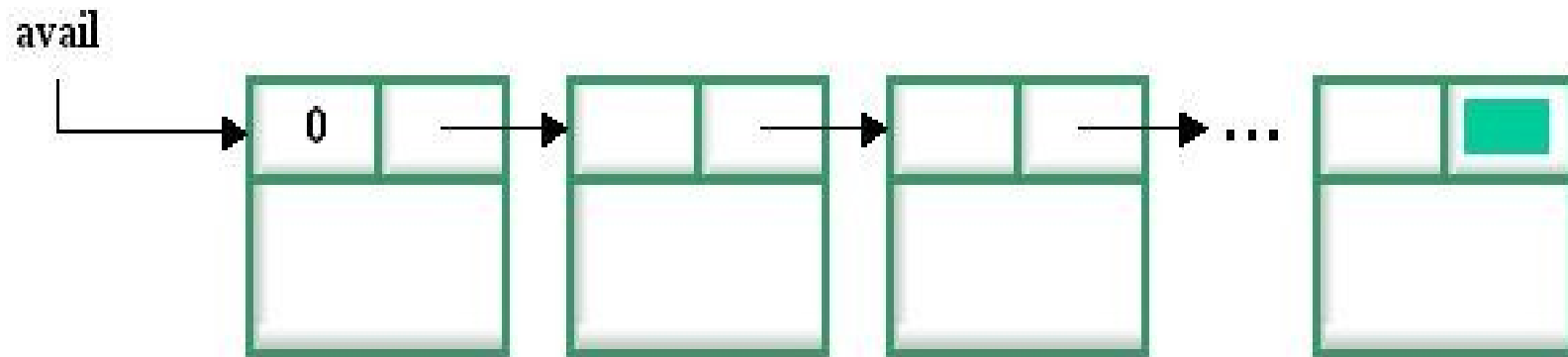
Alocação Dinâmica de Memória

- DMA – Gerenciamento do **memory pool** usando técnicas para alocação e desalocação de blocos
- É também conhecida como **alocação dinâmica de armazenamento**
- Envolve duas operações:
 - **Reserva** – alocação de memória para um pedido de tarefa
 - **Liberação** – retorna a memória antes alocada para o *memory pool*
- Duas técnicas:
 - Método ***sequential fit*** – todos os blocos liberados são concebidos em uma lista *singly-linked* chamada de lista disponível
 - Método ***buddy-system*** - blocos são alocados apenas em tamanhos *quantum*



Alocação Dinâmica de Memória: *Sequential-Fit*

- Reserva

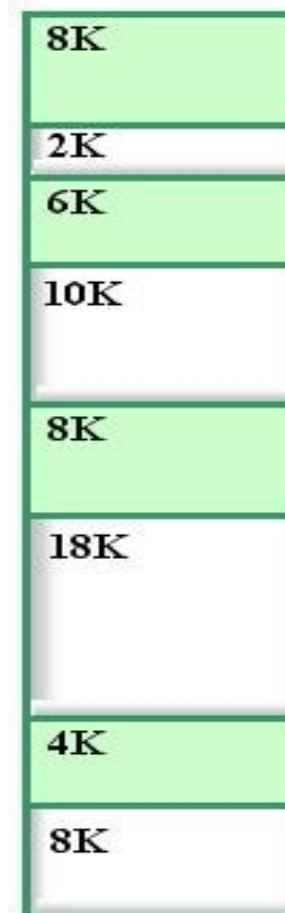


Alocação Dinâmica de Memória: *Sequential-Fit*

- Reserva

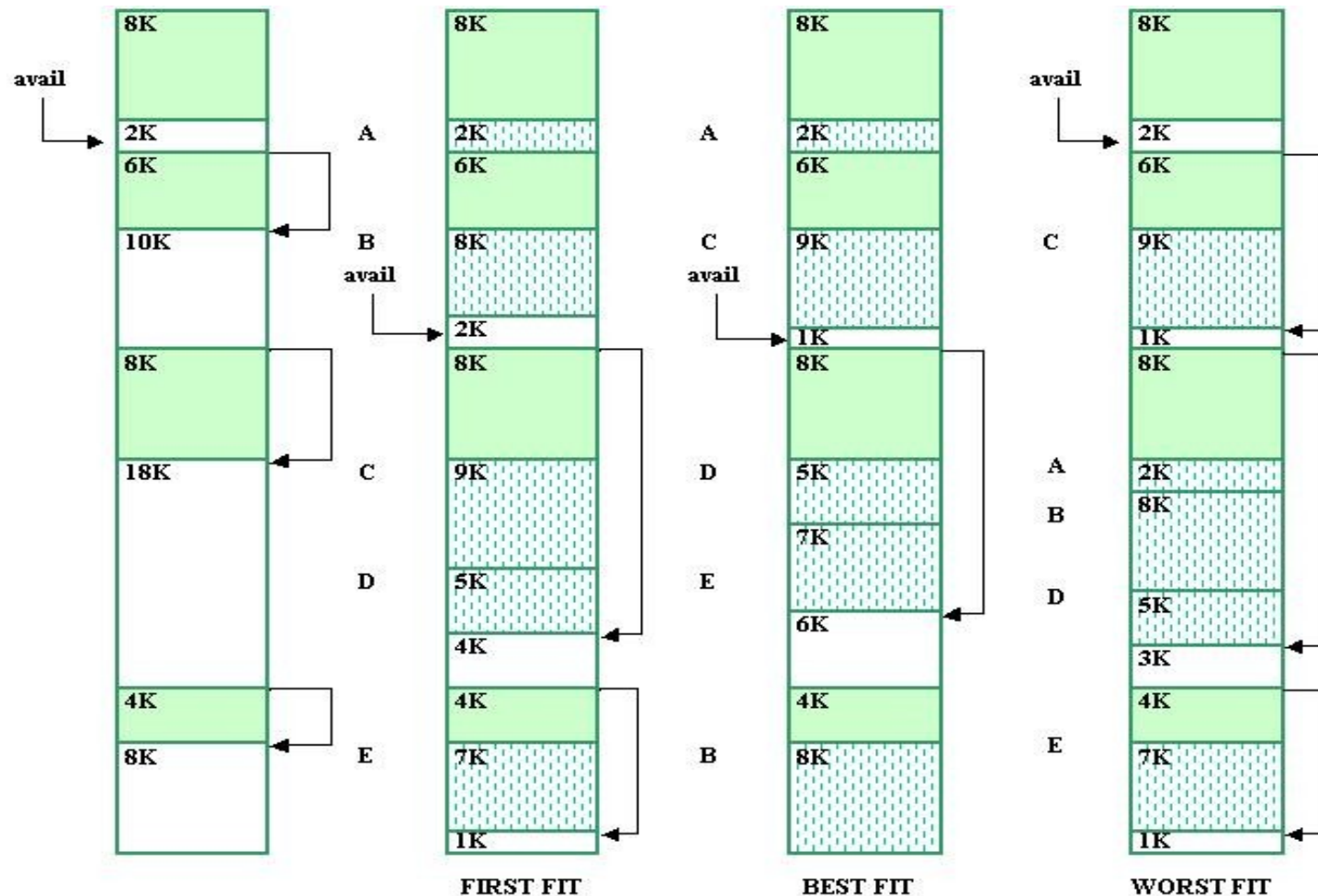
Tarefa	Pedido
A	2K
B	8K
C	9K
D	5K
E	7K

Memory Pool



Alocação Dinâmica de Memória: *Sequential-Fit*

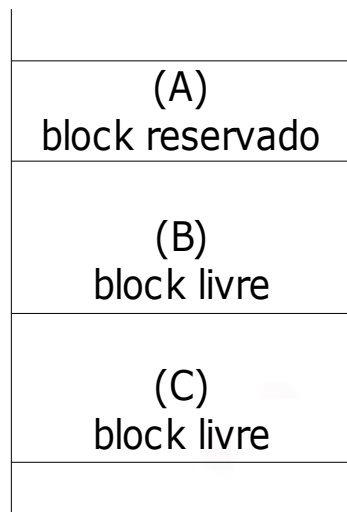
- Reserva



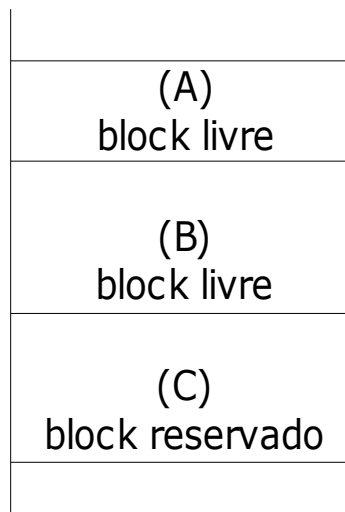
Estruturas de Dados

Alocação Dinâmica de Memória: *Sequential-Fit*

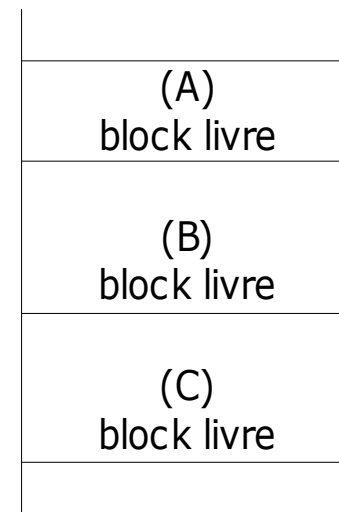
- Liberação



(b)



(c)

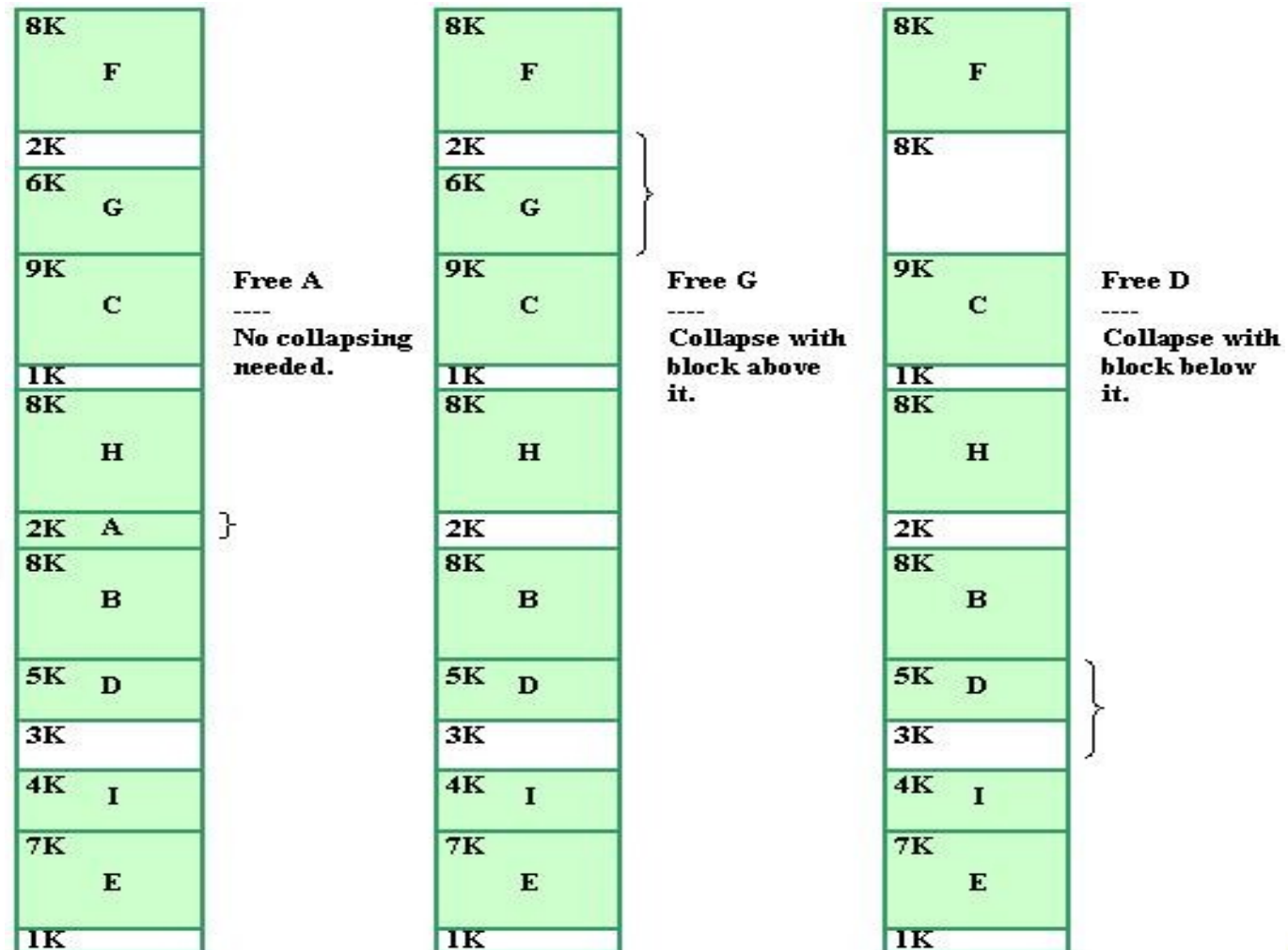


(d)

Possíveis casos na Liberação

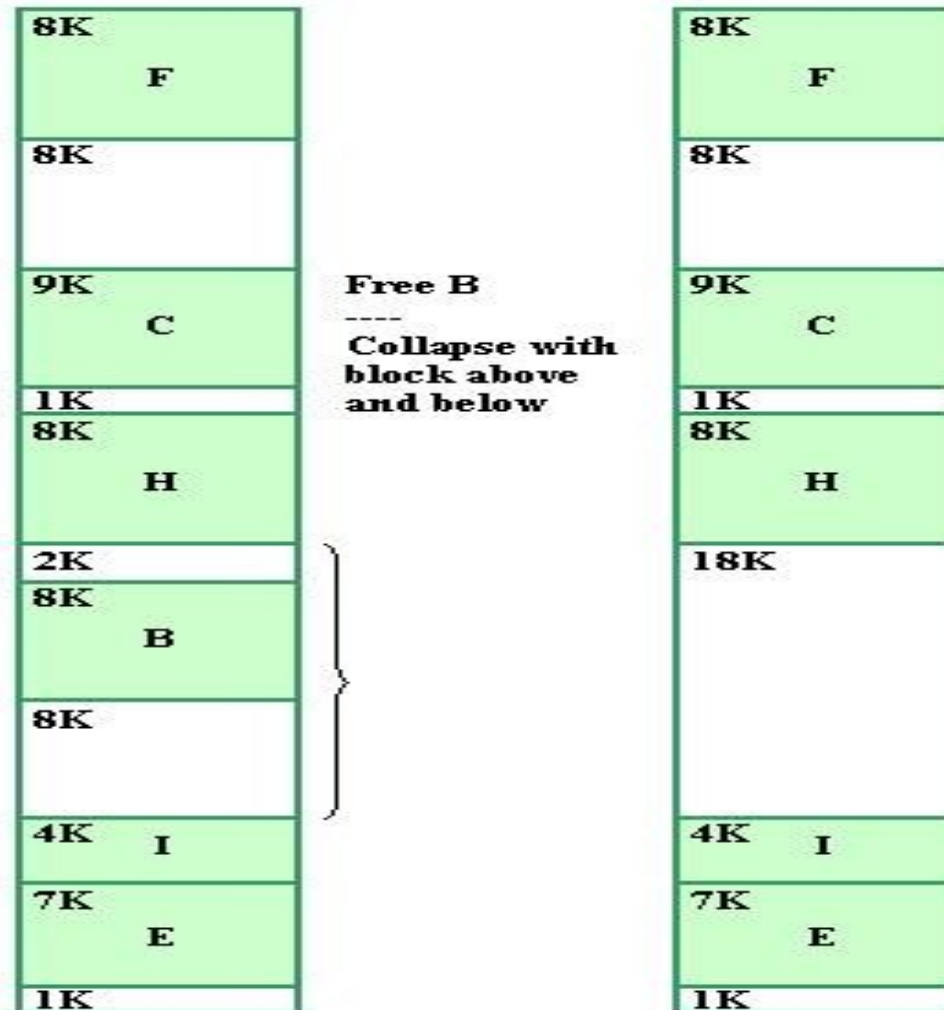
Alocação Dinâmica de Memória: *Sequential-Fit*

- Liberação
- *Sorted-List*



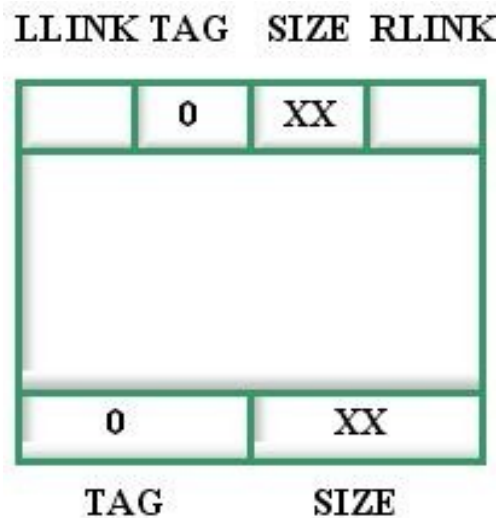
Alocação Dinâmica de Memória: *Sequential-Fit*

- Liberação
- *Sorted-List*

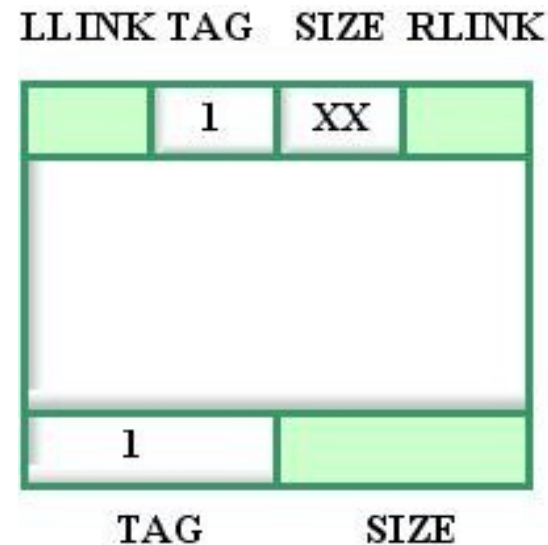


Alocação Dinâmica de Memória: *Sequential-Fit*

- Liberação
- *Boundary-Tag*



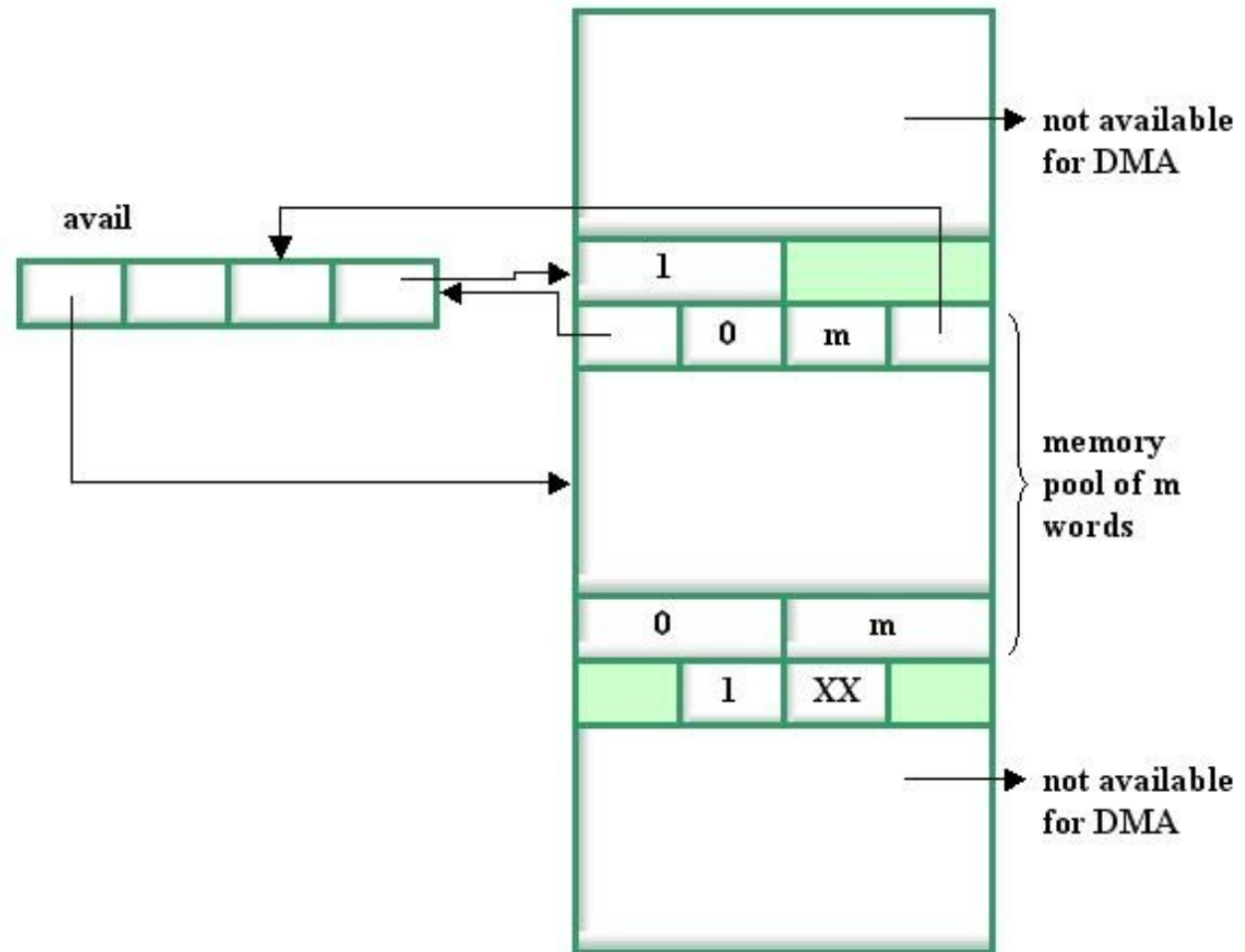
block livre



block reservado

Alocação Dinâmica de Memória: *Sequential-Fit*

- Liberação
- *Boundary-Tag*



Alocação Dinâmica de Memória: *Buddy-System*

- Blocos são alocados em tamanhos quantum
- Algumas listas disponíveis para cada tamanho alocável
- Dois métodos:
 - **Binary buddy-system** – os tamanhos dos blocos são baseados em potência de 2
 - **Fibonacci buddy-system** – os tamanhos dos blocos são baseados na série Fibonacci



Alocação Dinâmica de Memória: *Buddy-System*

- Método Binary Buddy-System: Reserva
- Método Binary Buddy-System: Liberação

Alocação Dinâmica de Memória: Fragmentação

- Fragmentação Externa
- Fragmentação interna

Sumário

- Definição e Conceitos Relacionados
 - Lista Generalizada
- Representações de Listas
 - Lista Circular *Singly-Linked*
 - Lista *Double-Linked*
- Aplicação: Aritmética Polinomial
- Alocação Dinâmica de Memória
 - *Sequential-Fit*
 - *Buddy-System*
 - *Fragmentação*



Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:

