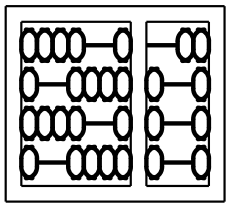


Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC

Estruturas ligadas

Listas ligadas

Definição: Uma lista ligada é um conjunto de itens em que cada item é representado por um nó contendo um *link* para outro nó.

Listas ligadas

Definição: Uma lista ligada é um conjunto de itens em que cada item é representado por um nó contendo um *link* para outro nó.

• Os nós de dados são ligados entre si para indicar a relação de ordem existente entre eles da seguinte forma: a cada item de dado é associada uma variável de ligação.

Listas ligadas

Definição: Uma lista ligada é um conjunto de itens em que cada item é representado por um nó contendo um *link* para outro nó.

• Os nós de dados são ligados entre si para indicar a relação de ordem existente entre eles da seguinte forma: a cada item de dado é associada uma variável de ligação.

• Para cada novo elemento a ser inserido na estrutura, **aloca-se** um espaço de memória para armazená-lo.

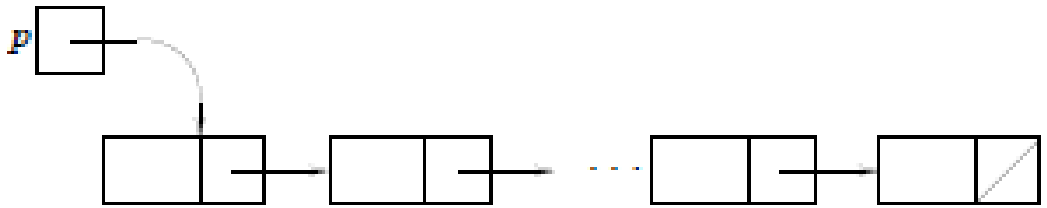
Listas ligadas

Definição: Uma lista ligada é um conjunto de itens em que cada item é representado por um nó contendo um *link* para outro nó.

• Os nós de dados são ligados entre si para indicar a relação de ordem existente entre eles da seguinte forma: a cada item de dado é associada uma variável de ligação.

• Para cada novo elemento a ser inserido na estrutura, **aloca-se** um espaço de memória para armazená-lo.

Listas ligadas simples



- Exemplo de declaração de um nó:

```
struct lista {
  int info ;              /* campo de dados */
  struct lista *prox ; /* auto-referencia */
} ;

typedef struct lista No, InicioLista, Lista, *TADLista ;
```

Exemplos de funções

- Exemplo de funções associadas ao TAD Lista:

```
/* funcao de criacao: RETORNA uma lista vazia */
Lista *lst_cria(void) {
 return NULL ;
}

/* insercao no inicio: RETORNA a lista atualizada */
Lista *lst_insere(Lista *l, int i) {
 Lista *novo = (Lista *)malloc(sizeof(Lista)) ;
 novo->info = i ;
 novo->prox = l ;
 return novo ;
}
```

Exemplos de funções

• O programa que importa as funções (endereços passados por valor):

```
int main(void) {

 Lista *l ;
 l = lst_cria() ;  /* inicializa lista */
 l = lst_insere(l, 23) ;  /* insere dado */
 l = lst_insere(l, 45) ;
 ...
 return 0 ;
}
```

Exemplos de funções

• Funções com endereços passados por referência (alterados dentro da própria função)

```
/* função de criação */
void lst_cria(Lista **l) {
 *l = NULL ;
}

/* insere no inicio */
void lst_insere(Lista **l, int i) {
 Lista *novo = (Lista *)malloc(sizeof(Lista)) ;
 novo->info = i ;
 novo->prox = *l ;
 *l = novo ;
}
```

Exemplos de funções

• O programa que importa as funções (endereços passados por referência):

```
...
Lista *l ;
lst_cria(&l) ;              /*inicializa lista com NULL */
lst_insere(&l, 23) ;        /* passa o end. do ponteiro */
lst_insere(&l, 45) ;
...
```

Exemplos de funções

- Função que percorre os elementos da lista.

```
/* funcao imprime */

void lst_imprime(Lista *l) {
 Lista *p ;
 for(p = l; p!=NULL; p = p->prox)
   printf(``info = %d\n'', p->info) ;
}
```

Exemplos de funções

- Função que verifica se a lista está vazia.

```
/* funcao testa lista vazia */

int lst_vazia(Lista *l) {
 return (l == NULL) ; /* retorna 1 se lista vazia e
                         0 em caso contrário */
}
```

Exemplos de funções

- Função que insere um nó entre um nó p e o seguinte.

```
/* insere entre o no p e o seguinte (supoe p != NULL) */

void  insere(No *p, int x) {
  No *novo ;
  novo = (No *)malloc(sizeof(No)) ;
  novo->info = x ;
  novo->prox = p->prox ;
  p->prox = novo ;
}
```

→ Não pode ser usada para inserir no início da lista.

Exemplos de funções

- Função que insere um nó entre um nó p e o seguinte.

```
/* insere entre o no p e o seguinte (supoe p != NULL) */

void  insere(No *p, int x) {
  No *novo ;
  novo = (No *)malloc(sizeof(No)) ;
  novo->info = x ;
  novo->prox = p->prox ;
  p->prox = novo ;
}
```

$\rightarrow$ Não pode ser usada para inserir no início da lista.

Exemplos de funções

- Função que recebe um nó p e remove o nó seguinte.

```
/* remove o no p->prox */

void remove(No *p) {
 No *temp ;
 temp = p->prox ;
 p->prox = temp->prox ;
 free(temp) ;
}
```

→ Não pode ser usada para remover o primeiro nó da lista.

Exemplos de funções

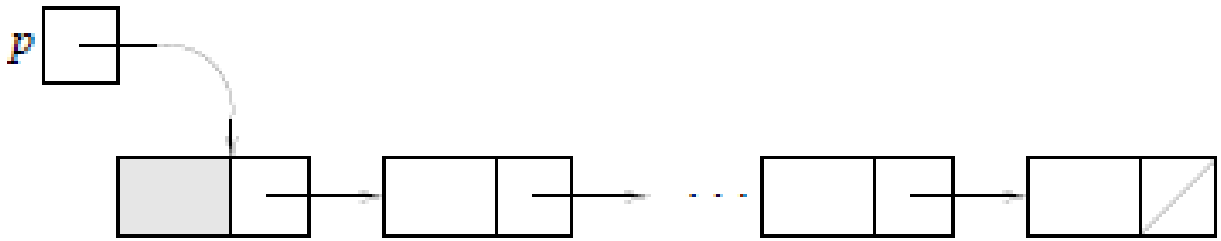
- Função que recebe um nó p e remove o nó seguinte.

```
/* remove o no p->prox */

void remove(No *p) {
 No *temp ;
 temp = p->prox ;
 p->prox = temp->prox ;
 free(temp) ;
}
```

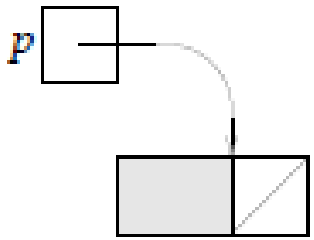
$\rightarrow$ Não pode ser usada para remover o primeiro nó da lista.

Exemplos de funções

- Listas com definição de um *nó cabeça* evitam os "problemas" (especificidades) anteriores.



Lista vazia:



Exemplos de funções

- Inicializa lista com nó cabeça.

```
void InicializaLista(InicioLista *inicio) {

inicio->info = -1 ;      /* valor diferenciado */
inicio->prox = NULL ;
}
```

Exemplos de funções

- Insere no início da lista com nó cabeça.

```
void InsereInicio(InicioLista *inicio, int dado) {

 No *temp ;

 temp=(No *)malloc(sizeof(No)) ;

 temp->info = dado ;
 temp->prox = inicio->prox ;

 inicio->prox = temp;
}
```

Exemplos de funções

- Imprime lista com nó cabeça.

```
void ImprimeLista(InicioLista *inicio) {

 No *temp ;

   for(temp = inicio->prox; temp != NULL; temp = temp->prox)
   printf("%d\n", temp->info) ;
}
```

Exemplos de funções

- Exemplo de programa que importa as funções com nó cabeça

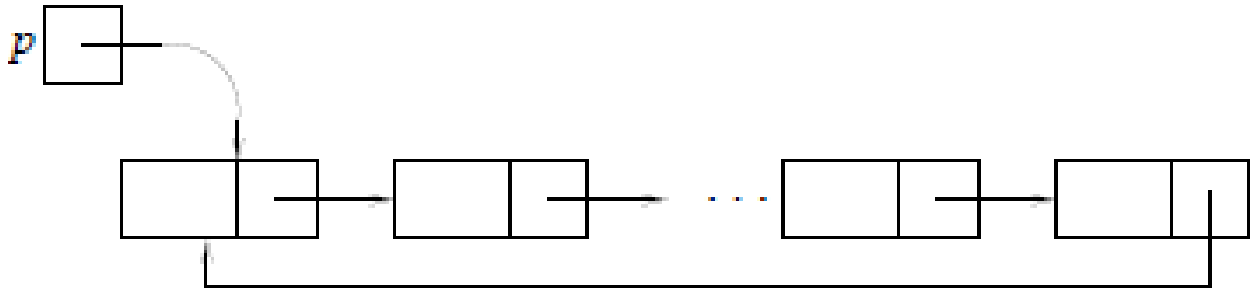
```
int main() {
 No ApLista ;
 int dado ;
 char sn[4]  ;
 InicializaLista(&ApLista) ;      /* Inicializa no cabeca */
 do {                             /* insere no inicio */
  printf("Digite inteiro a ser inserido na lista\n");
  scanf("%d", &dado) ;

  InsereInicio(&ApLista, dado) ;

  printf("Mais um? (sim ou nao)?\n") ;
  scanf("%s",sn) ;
 }
 while(strcmp(sn,"nao")) ;
```

Exemplos de funções

- ... continuação

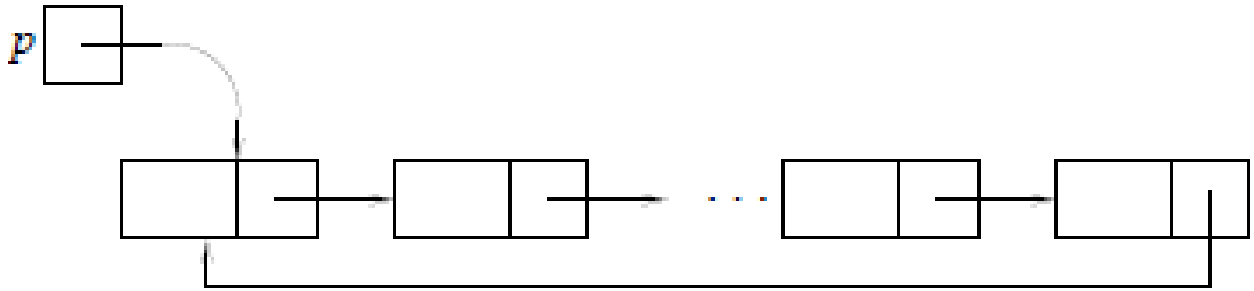
```
...
/* exibir conteudo da lista */

 ImprimeLista(&ApLista) ;
 return 0;
}
```

- Lista simples circular



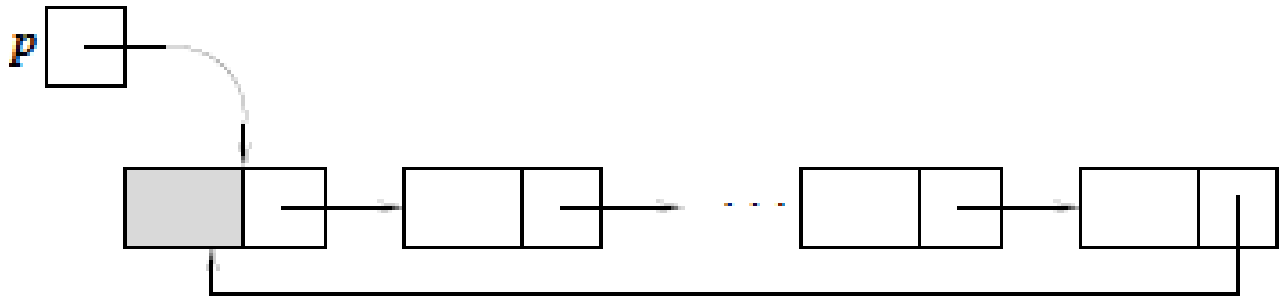
$\Rightarrow$ Problema: representação de lista vazia

- Lista simples circular

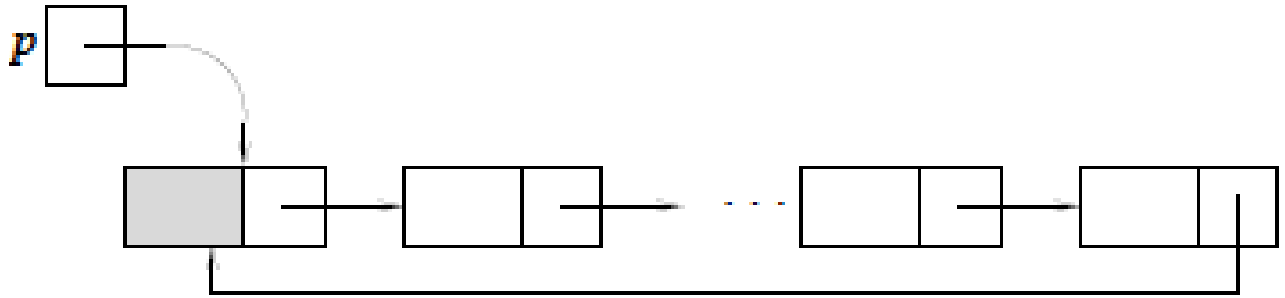


$\Rightarrow$ Problema: representação de lista vazia

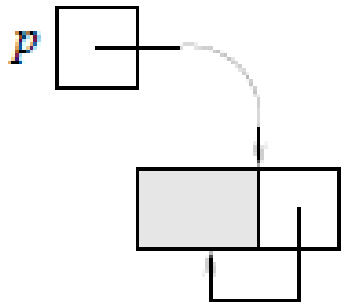
- Lista circular com nó cabeça



- Lista circular com nó cabeça



- Representação de lista vazia:



p->prox = p ;

Exemplos de funções

- Insere dado numa lista circular (com nó cabeça)

```
void InsereCircular(InicioLista *inicio, int x) {

 No *temp = (No *)malloc(sizeof(No)) ;
 temp->info = x ;
 temp->prox = inicio->prox ;
 inicio->prox = temp ;
}
```

Exercício: Defina a função que insere dado numa lista circular sem nó cabeça.

Exemplos de funções

- Insere dado numa lista circular (com nó cabeça)

```
void InsereCircular(InicioLista *inicio, int x) {

 No *temp = (No *)malloc(sizeof(No)) ;
 temp->info = x ;
 temp->prox = inicio->prox ;
 inicio->prox = temp ;
}
```

Exercício: Defina a função que insere dado numa lista circular sem nó cabeça.

Exemplos de funções

- Busca em uma lista circular com *sentinela* (o nó cabeça indicará o final da lista)

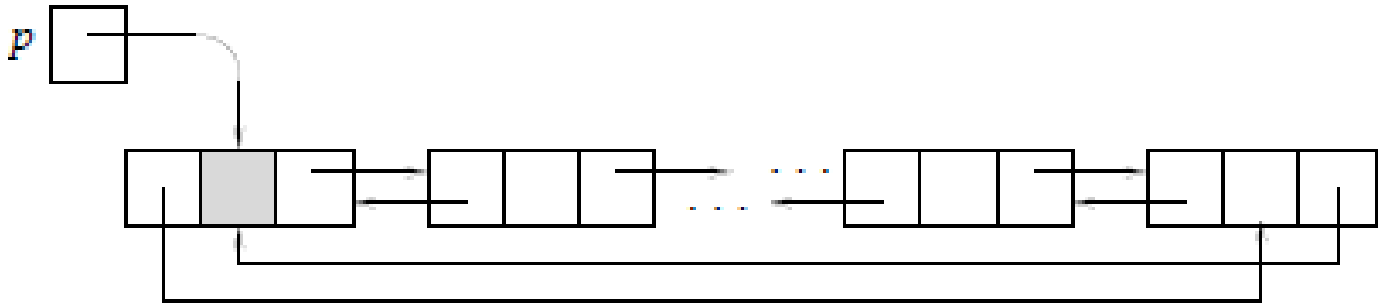
```
No *BuscaCircular(InicioLista *inicio, int x) {

 No *temp = inicio ;
 inicio->info = x ;   /* sentinela */
 do {
   temp = temp->prox ;
 } while(temp->info != x) ;

 if(temp == inicio) return NULL ;   /* nao encontrou x */
 else return temp ;
}
```

Exemplos de funções

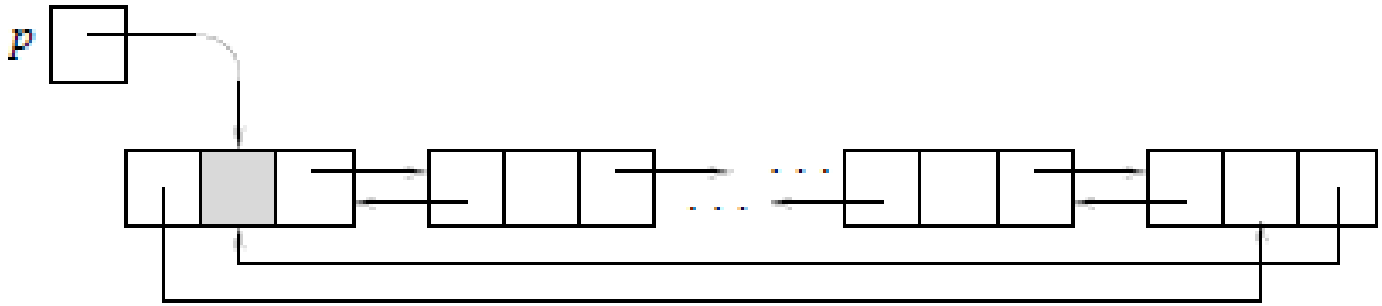
- Busca em uma lista circular sem *sentinela*

```
No *BuscaCircular(InicioLista *inicio, int x) {

 No *temp = inicio ;

 do {
   temp = temp->prox ;
 } while((temp != inicio) && (temp->info != x)) ;

 if(temp == inicio) return NULL ;  /* nao encontrou x */
 else return temp ;
}
```

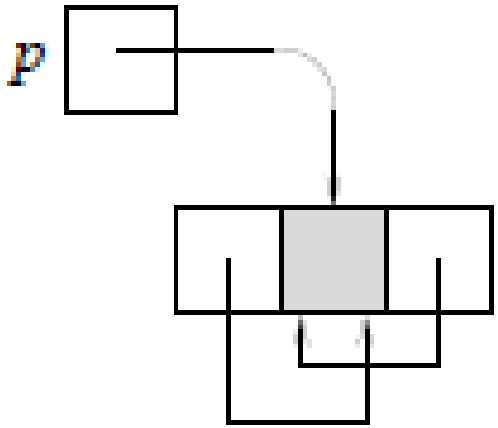
- Lista duplamente ligada com nó cabeça



- Lista duplamente ligada com nó cabeça



- Representação de lista vazia:



p->esq = p ; p->dir = p ;

- Exemplo de declaração de um nó

```
typedef struct ListaDupla {
 int info ;
 struct ListaDupla *esq, *dir ;
} No, InicioLista, ListaDupla ;
```

Exemplos de funções

- Insere dado à direita

```
void InsereDuplaDireita(ListaDupla *inicio, int x) {

 No *temp = (No *)malloc(sizeof(No)) ;
 temp->dir = inicio->dir ;
 temp->esq = inicio ;
 inicio->dir->esq = temp ;
 inicio->dir = temp ;
 temp->info = x ;
}
```

- Exercício: Implementar InsereDuplaEsquerda

Exemplos de funções

- Insere dado à direita

```
void InsereDuplaDireita(ListaDupla *inicio, int x) {

 No *temp = (No *)malloc(sizeof(No)) ;
 temp->dir = inicio->dir ;
 temp->esq = inicio ;
 inicio->dir->esq = temp ;
 inicio->dir = temp ;
 temp->info = x ;
}
```

- Exercício: Implementar InsereDuplaEsquerda

Exemplos de funções

- Remove um nó da lista duplamente ligada

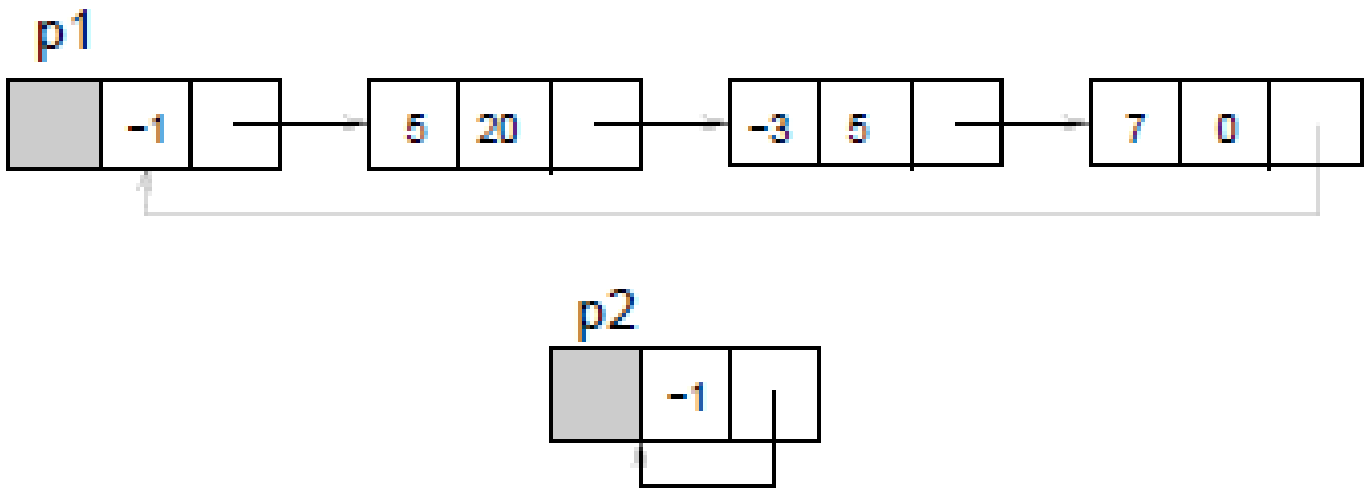
```
void RemoveDupla(ListaDupla *p) {

  p->esq->dir = p->dir ;
  p->dir->esq = p->esq ;
  free(p) ;
}
```

Exemplos de Aplicação

- Operações com polinômios

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 x^0$$

Representação ligada de $P_1(x) = 5x^{20} - 3x^5 + 7$ e $P_2(x) = 0$:



- Declaração da estrutura para polinômios

```
typedef struct TermoPol {
 int expo ;
 float coef ;
 struct TermoPol *prox ;
} Termo, *Polinomio, *ApTermo ;
```

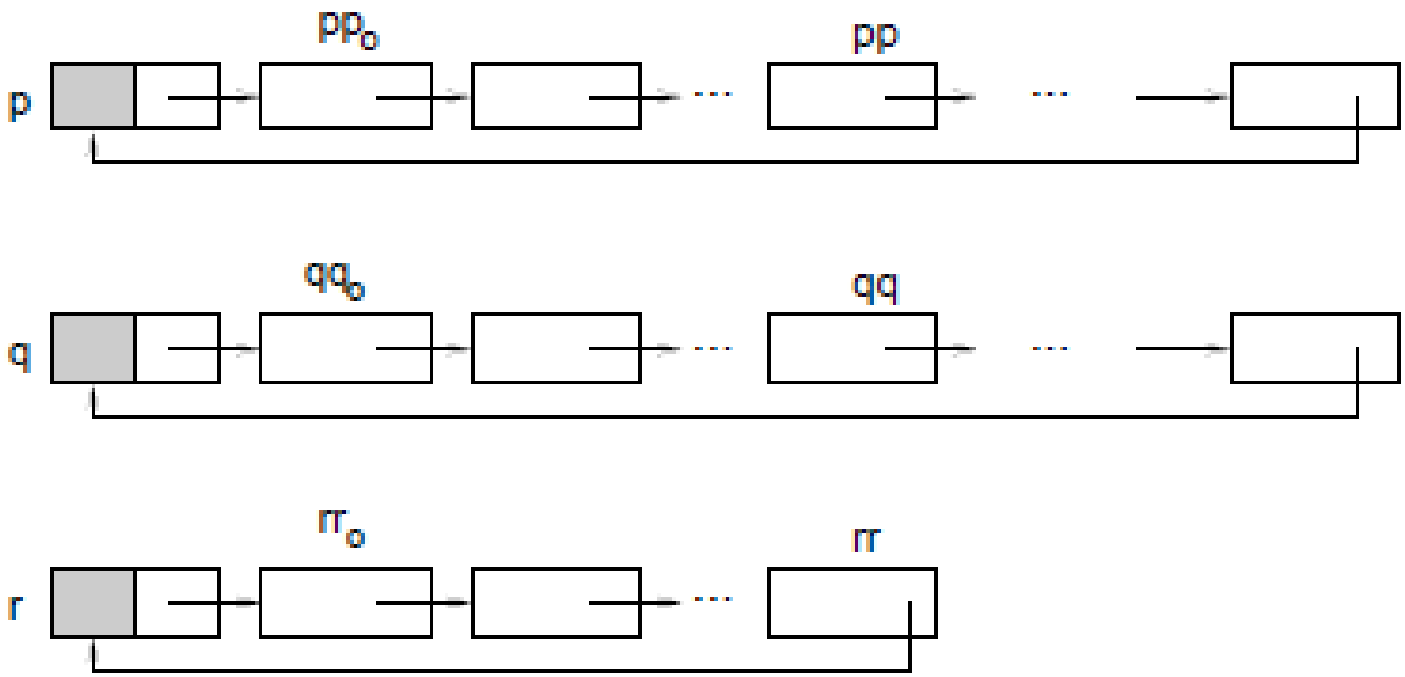
Exemplos de funções

- Imprime polinômio

```
void ImprimePolinomio(Polinomio p) {
 if(p->prox == p) {
  printf(``Polinomio nulo\n'') ;
  return ;
 }

 p = p->prox ;
 while(p->expo != -1) {
  printf(``(%2d, %5.1f)  ``, p->expo, p->coef) ;
  p = p->prox ;
 }
 printf(``\n'') ;
}
```

Exercício

- Soma de polinômios:

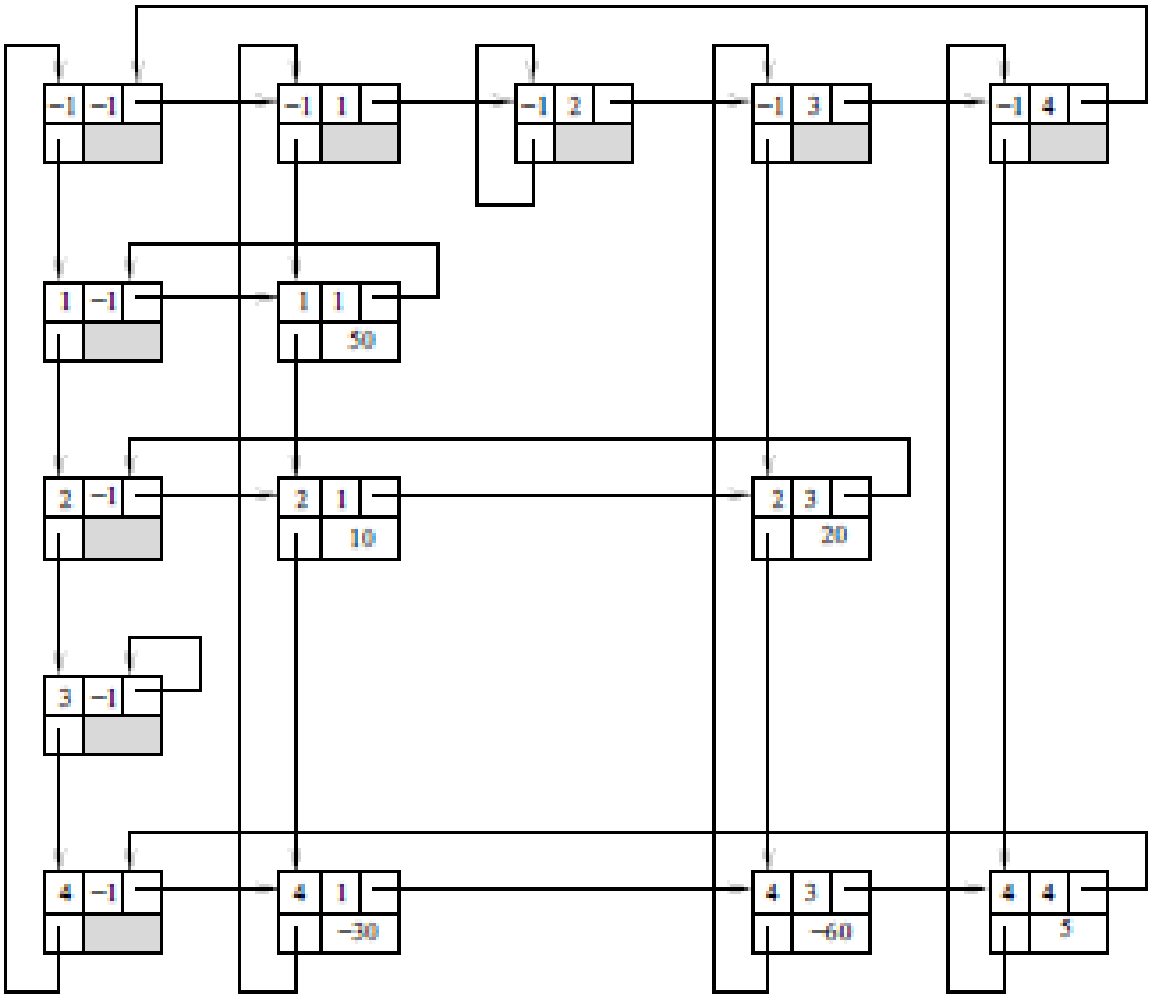


Outro exemplo de aplicação

• Representação de matriz esparsa por uma lista ligada circular com nó cabeça

$$\begin{vmatrix} 50 & 0 & 0 & 0 \\ 10 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 \\ -30 & 0 & -60 & 5 \end{vmatrix}$$

- Implementação: listas ortogonais



- Declaração de um nó da lista

```
typedef struct No {
 int lin, col ;
 float valor ;
 struct No *baixo, *direita ;
} NoMatriz, *MatrizEsparsa ;
```

- Exercício: Implementar um TAD MatrizEsparsa que executa as funções *InicializaMatriz, LiberaMatriz, LeElementoMatriz, AtribuiElemMatriz, SomaMatrizes, MultiplicaMatrizes*.

• Declaração de um nó da lista

```
typedef struct No {
 int lin, col ;
 float valor ;
 struct No *baixo, *direita ;
} NoMatriz, *MatrizEsparsa ;
```

• Exercício: Implementar um TAD MatrizEsparsa que executa as funções *InicializaMatriz*, *LiberaMatriz*, *LeElementoMatriz*, *AtribuiElemMatriz*, *SomaMatrizes*, *MultiplicaMatrizes*.

Exemplos de funções

- Inserir valor na matriz esparsa

```
void InsereMatriz(MatrizEsparsa p, int i, int j, float v) {

NoMatriz *pi, *pj ; /* predecessores na linha e coluna */
NoMatriz *r ;
int k ;

pi= p ;       /* visitar nos cabeças das linhas */
for(k=1; k<=i; k++)
 pi = pi->baixo ;    /* pi aponta para o predecessor
                        na linha i */

 while((pi->direita->col != -1) && (pi->direita->col < j))
  pi = pi->direita    /* segue a direita */
```

```
 pj = p ;  /* visitar nos cabeças das colunas */
 for(k=1; k<=j; k++)
  pj = pj->direita ;   /* pj aponta para o predecessor
                          na coluna j */

 while((pj->baixo->lin != -1) && (pj->baixo->lin < i ))
  pj = pj -> baixo ;    /* segue para baixo */

 /* insercao */
 r = (NoMatriz)malloc(sizeof(NoMatriz)) ;
 r->lin = i;  r->col = j;  r->valor = v ;
 r->direita = pi->direita ;
 r->baixo = pj->baixo ;
 pi->direita = r ;
 pj->baixo = r ;
}  /* fim da funçao */
```