

Implementação com array

```
void moveToStart(List l) void moveToEnd(List l)
l.curr ← 0;               l.curr ← l.listSize;

void prev(List l)
l if l.curr ≠ 0 then l.curr--;

void next(List l)
l if l.curr < l.listSize then curr++;
```

Operações implementadas

Mesma implementação, independentemente da estrutura de dados.

Tipo composto:

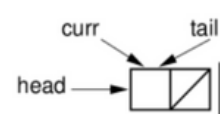
- 1 int maxSize; //capacidade
- 2 int tamanho da lista; //número de elementos
- 3 int atual; // posição do cursor
- 4 E[] listaArray; //array com valores

Operações da lista

```
void clear(List l);
void insert(List l, E item);
void append(List l, E item);
E remove(List l);
void moveToStart(List l);
void moveToEnd(List l);
void prev(List l);
void next(List l);
int length(List l);
int currPos(List l);
void moveToPos(List l, int pos);
E getValue(List l);
```

Listar ADT como uma lista vinculada

Lista vazia: nó de cabeçalho vs. casos de espaço (insert and remove)



Atenção para onde curr está apontando (não como segue)

Tipo composto (List):

- 1 Link head;
- 2 Link tail;
- 3 Link curr;
- 4 int cnt;

Tipo composto (Link/nó):

- 1 elemento E; // valor armazenado neste link/nó
- 2 Link próximo; // referência ao próximo link/nó

Algoritmo: Link criar link (E it, Link nextval)

l n.elemento ← isso; n.próximo ← próximo valor; retornar n;

Algoritmo: Link criar link(Link nextval)

l n.próximo ← próximo valor; retornar n;

LIST ADT

Lista de ADT

Abstração matemática
Sequências de elementos homogêneos
Um símbolo l para a posição atual do cursor. Assim podendo inserir ou remover um elemento da lista.

Tipos de dados abstratos e estrutura de dados

Tipo: uma coleção de valores
Tipo simples vs. tipo composto
Tipo de dados: um tipo e operações para manipular o tipo
Tipo de dados abstrato (ADT): um tipo de dados como componente de software
Estrutura de dados: implementação para um ADT
Várias implementações para o mesmo ADT

ADT(lógica) -> Operações
Data Structure (física) -> armazenamento de espaço, subroutines

Outras informações

Listar ADT como uma matriz

- ± Tamanho máximo predefinido (alternativa: matrizes dinâmicas)
 - A alocação dinâmica de matrizes não implica matrizes dinâmicas
 - Análise amortizada (anexo): custo da cópia deixa de ser relevante
- + Sem espaço extra com ponteiros (links)
- Espaço consumido por posições não utilizadas

Listar ADT como uma lista vinculada

- + Sem tamanho máximo predefinido
- + Sem espaço extra para elementos que não estão mais acessíveis
- Espaço consumido por ponteiros (links)

Variações

Conjunto de nós livres (lista livre)

Listas duplamente vinculadas

Listas vinculadas circularmente