

Lista Estática Sequencial

Prof^a.Dr^a.Thatyana de Faria Piola Seraphim
Prof.Dr. Enzo Seraphim

Universidade Federal de Itajubá

thatyana@unifei.edu.br
seraphim@unifei.edu.br

Definição de Lista

Uma lista é uma estrutura que armazena elementos de forma alinhada, ou seja, os elementos são dispostos um após o outro, como em uma lista de nomes, peças, valores, pessoas, compras.

- ▶ Uma lista nada mais é do que uma sequência de zero ou mais elementos x_1, x_2, \dots, x_n , onde:
 - ▶ x_i é de um determinado tamanho;
 - ▶ **n** representa o tamanho da lista.

Introdução

Definição de Lista

A principal propriedade estrutural de uma lista envolve as posições dos itens em uma dimensão. Assumindo $n \geq 1$:

- ▶ x_1 é o primeiro elemento da lista e x_n é o último elemento da lista;
- ▶ x_i é o elemento antecessor de x_{i+1} para $i = 1, 2, \dots, n - 1$;
- ▶ x_i é o elemento sucessor de x_{i-1} para $i = 2, 3, \dots, n$;
- ▶ é dito que o elemento x_i está na i -ésima posição da lista.

Uma lista pode ser implementada de duas formas:

- ▶ **Estática.**
- ▶ **Dinâmica.**

Estática:

- ▶ Todo o espaço de memória a ser utilizado (para armazenar os itens) é reservado (alocado) no início da execução do programa.
- ▶ Esse espaço de memória permanece reservado durante toda a execução do programa, independente de estar sendo efetivamente utilizado ou não.
- ▶ Os elementos podem ser dispostos de duas formas:
 1. **Sequencial**: os itens ficam, necessariamente, em sequência (um ao lado do outro) na memória.
 2. **Encadeada**: os itens não estão, necessariamente, em posições de memória adjacentes, ou seja, em posições próximas umas das outras.
- ▶ As listas estáticas são implementadas utilizando um **vetor**.

Dinâmica:

- ▶ O espaço de memória a ser utilizado (para armazenar os itens) pode ser reservado (alocado) no decorrer da execução de um programa, quando for efetivamente necessário.
- ▶ O espaço reservado pode ser liberado durante a execução do programa, quando não for mais necessário.
- ▶ As listas dinâmicas são implementadas utilizando **ponteiros e variáveis dinâmicas**.

Lista Estática Sequencial:

- ▶ É um arranjo de registros onde estão estabelecidos regras de precedência entre seus elementos, ou seja, é uma coleção ordenada de componentes do mesmo tipo.
- ▶ O sucessor de um elemento ocupa posição física subsequente.
 - ▶ Por exemplo: lista telefônica, lista de alunos.
- ▶ A implementação de operações pode ser feita utilizando **vetor**,
 - ▶ onde o vetor associa o elemento **a[i]** com o índice **i** (mapeamento sequencial).

Lista Estática Sequencial

Características

- ▶ Os elementos da lista são armazenados ordenados e em posições consecutivas (sequenciais) de memória.
- ▶ A lista pode ser percorrida em qualquer direção.
- ▶ A inserção de um novo elemento pode ser realizada após o último elemento com custo constante.
- ▶ A inserção de um novo elemento no meio da lista requer um deslocamento de todos os elementos localizados após o ponto de inserção.
 - ▶ A inserção e o deslocamento dos elementos tem um custo linear, ou seja, uma pequena quantidade de trabalho é realizada sobre cada elemento.

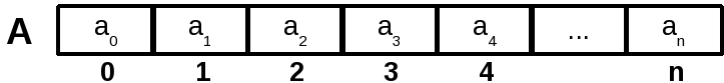
Lista Estática Sequencial

Características

- ▶ A retirada de um elemento do início da lista requer um deslocamento de elementos para preencher o espaço deixado vazio.
 - ▶ Por exemplo: para remover o elemento **$a[i]$** é necessário deslocar à esquerda do elemento **$a[i+1]$** até o último.
 - ▶ A remoção dos elementos da lista também tem um custo linear.

Lista Estática Sequencial

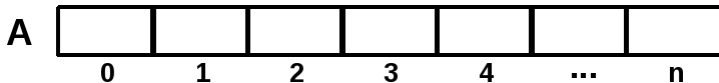
Caraterísitcas



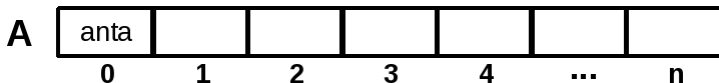
- Inserir os elementos: anta, gato, cabra, boi

Lista Estática Sequencial

Caraterísitcas

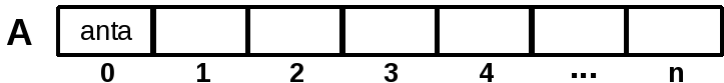


- Inserir os elementos: **anta**, gato, cabra, boi

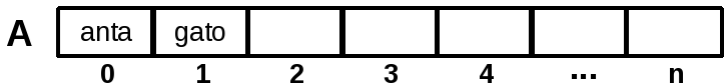


Lista Estática Sequencial

Caraterísitcas

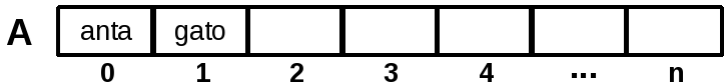


- Inserir os elementos: anta, gato, cabra, boi

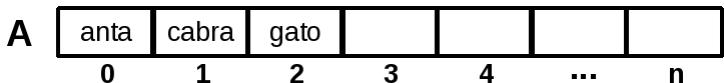


Lista Estática Sequencial

Caraterísitcas

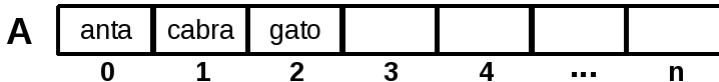


- Inserir os elementos: anta, gato, **cabra**, boi

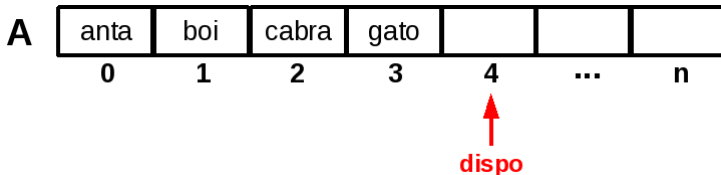


Lista Estática Sequencial

Caraterísitcas



- Inserir os elementos: anta, gato, cabra, **boi**



Lista Estática Sequencial

Propriedades

As propriedades da lista estática permitem responder questões como:

1. Qual é o primeiro elemento da lista?
 2. Qual é o último elemento da lista?
 3. Quais elementos sucedem um elemento?
 4. Quantos elementos existem na lista?
 5. Inserir um elemento na lista.
 6. Eliminar um elemento da lista.
- ▶ As 4 primeiras questões são feitas em tempo constante.
 - ▶ As operações de inserção e remoção precisam ser tratadas com um pouco mais de cuidado.

Lista Estática Sequencial

Propriedades

O conjunto de operações que podem ser realizadas na lista depende de cada aplicação.

1. Criar uma lista linear vazia.
2. Inserir um novo elemento imediatamente após o i -ésimo elemento.
3. Retirar o i -ésimo elemento da lista.
4. Localizar o i -ésimo elemento para examinar e/ou alterar o conteúdo de seus componentes.
5. Combinar duas ou mais listas em uma única lista.
6. Partir uma lista em duas ou mais listas.
7. Fazer uma cópia da lista.
8. Ordenar os elementos da lista em ordem ascendente ou descendente.
9. Pesquisar a ocorrência de um elemento com um valor particular.

Lista Estática Sequencial

Vantagens/Desvantagens/Uso

Vantagens:

- ▶ Acesso direto a qualquer elemento da lista.
- ▶ Tempo constante para acessar o i -ésimo elemento (dependerá somente do índice).

Desvantagens:

- ▶ Movimentação quando o item é eliminado/inserido.
- ▶ Tamanho máximo pré-estimado.

Quando usar:

- ▶ Listas pequenas.
- ▶ Inserção/remoção no fim da lista.
- ▶ Tamanho máximo bem definido.

Operações da Lista Sequencial

```
1  #include<stdio.h>
2  #define MAX 100 //quantidade maxima de elementos na lista
3  typedef enum{false, true} bool; //tipo booleano
4  int listaEstSeq[MAX]; //lista sequencial
5  int dispo=0; //proxima posicao disponivel na lista
6
7  //Operacoes lista estatica sequencial
8  int primeiroListaEstSeq(); //retorna primeiro elemento
9  int ultimoListaEstSeq(); //retorna ultimo elemento
10 //retorna a quantidade de elementos da lista
11 int quantListaEstSeq();
```

Cont. Operações da Lista Sequencial

```
12  //retorna verdadeiro se inseriu o elemento
13  bool insereListaEstSeq(int valor);
14  //retorna verdadeiro se removeu o elemento
15  bool removeListaEstSeq(int valor);
16  //imprime na tela os elementos da lista
17  void imprimeListaEstSeq();
18  //retorna a posicao de um elemento
19  int pesqSeqListaEstSeq(int chave);
20  //retorna a posicao de um elementos
21  int pesqBinListaEstSeq(int chave);
```

Lista Estática Sequencial

Algoritmos

Função main()

```
1  int main(int argc, char **argv){
2      int aux, i;
3      //insere elemento
4      for(i=0; i<MAX; i++){
5          aux = rand() % (MAX*2);
6          if(pesqSeqListaEstSeq(aux) == -1){
7              insereListaEstSeq(aux);
8          }//end if
9          else{
10             i--;
11         }//end else
12     }//end for
```

Lista Estática Sequencial

Algoritmos

cont. Função main()

```
13     imprimeListaEstSeq();
14     printf("Valor procurado: ");
15     scanf("%d", &aux);
16     printf("Encontrado na posicao %d\n",
17           pesqSeqListaEstSeq(aux));
18     removeListaEstSeq(aux);
19     imprimeListaEstSeq();
20     return 0;
21 } //end main
```

Lista Estática Sequencial

Algoritmos

Primeiro elemento da lista

```
1  //retorna o primeiro elemento ou  
2  //-1 se a lista estiver vazia  
3  int primeiroListaEstSeq(){  
4      if(dispo == 0)  
5          return -1;  
6      else  
7          return listaEstSeq[0];  
8  } //end primeiroListaEstSeq
```

Lista Estática Sequencial

Algoritmos

Último elemento da lista

```
1  //retorna o ultimo elemento ou  
2  // -1 se a lista estiver vazia  
3  int ultimoListaEstSeq(){  
4      if(dispo == 0)  
5          return -1;  
6      else  
7          return listaEstSeq[dispo-1];  
8  } //end ultimoListaEstSeq
```

Lista Estática Sequencial

Algoritmos

Quantidade de elementos da lista

```
1  //retorna a quantidade de elementos da lista
2  int quantListaEstSeq(){
3      return dispo;
4  }//end quantListaEstSeq
```

Lista Estática Sequencial

Algoritmos

Impressão dos elementos da lista

```
1  //imprime na tela os elementos da lista
2  void imprimeListaEstSeq(){
3      int i = 0;
4      while(i < dispo){
5          printf("[(%2d)=%3d] ", i, listaEstSeq[i]);
6          i++;
7          if((i%10) == 0){ //pula linha a cada 10 linhas
8              printf("\n");
9          } //end if
10     } //end while
11     printf("\n");
12 } //end imprimeListaEstSeq
```


Lista Estática Sequencial

Algoritmos

Inserção de um elemento na lista

```
1  bool insereListaEstSeq(int valor){
2      int i = dispo;
3      if(dispo > MAX){
4          return false;
5      }//end if
6      else{
7          while((i > 0) && (listaEstSeq[i-1] > valor)){
8              listaEstSeq[i] = listaEstSeq[i-1];
9              i--;
10         }//end while
11         listaEstSeq[i] = valor;
12         dispo++;
13         return true;
14     }//end else
15 }//end insereListaEstSeq
```

Lista Estática Sequencial

Algoritmos

Remoção de um elemento na Lista

```
1  //retorna verdadeiro se removeu o elemento na lista
2  bool removeListaEstSeq(int valor){
3      int i = 0;
4      while((i < dispo) && (listaEstSeq[i] < valor)){
5          i++;
6      }//end while
7      if((i < dispo) && (listaEstSeq[i] == valor)){
8          //se encontrou elemento
```

Lista Estática Sequencial

Algoritmos

Remoção de um elemento na Lista

```
9      while(i < dispo-1){
10          listaEstSeq[i] = listaEstSeq[i+1];
11          i++;
12      }//end while
13      dispo--;
14      return true;
15  }//end if
16  else{
17      return false;
18  }//end else
19 }//end removeListaEstSeq
```

Lista Estática Sequencial

Algoritmos

Pesquisa Sequencial

```
1  //retorna a posicao de um elemento
2  //ou -1 caso nao encontrou
3  int pesqSeqListaEstSeq(int chave){
4      int i = 0;
5      while((i < dispo) && (listaEstSeq[i] < chave)){
6          i++;
7      }//end while
8      if((i < dispo) && (listaEstSeq[i] == chave)){
9          return i;
10     }//end if
11     else{
12         return -1;
13     }//end else
14 }//end pesqSeqListaEstSeq
```

Lista Estática Sequencial

Pesquisa Binária

Pesquisa ou Busca Binária (*Binary Search Algorithm*):

- ▶ É um algoritmo que busca um valor em um vetor ordenado.
- ▶ O algoritmo realiza sucessivas divisões no vetor (divisão e conquista) comparando o elemento a ser buscado (chave) com o elemento no meio do vetor.
 - ▶ Se o elemento do meio do vetor for a chave, a busca termina com sucesso.
 - ▶ Se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor.
 - ▶ Se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.
- ▶ Vantagem é que esta técnica é mais rápida para encontrar um elemento do que a pesquisa sequencial.

Lista Estática Sequencial

Pesquisa Binária – Algoritmo

Pesquisa Binária

```
1  //retorna a posicao de um elemento ou  
2  // -1 caso nao encontrou  
3  int pesqBinListaEstSeq(int chave){  
4      int meio, inf=0; //limite inferior  
5      int sup = MAX-1; //limite superior  
6      while(inf <= sup){  
7          meio=(inf + sup)/2;  
8          if(chave == listaEstSeq[meio]){  
9              return meio; //chave encontrada  
10         } //end if  
11         else{
```

Lista Estática Sequencial

Pesquisa Binária – Algoritmo

Cont. Pesquisa Binária

```
12         if(chave < listaEstSeq[meio]){
13             sup = meio-1; //chave esta na parte esquerda
14         }//end if
15         else{
16             inf = meio+1; //chave esta na parte direita
17         }//end else
18     }//end else
19 }//end while
20 return -1; // nao encontrado
21 }//end pesBinListaEstSeq
```

Lista Estática Sequencial

Pesquisa Binária

0	1	2	3	4	5	6	7	8	9
aguia	anta	bode	boi	cabra	cobra	coelho	gato	pato	rato

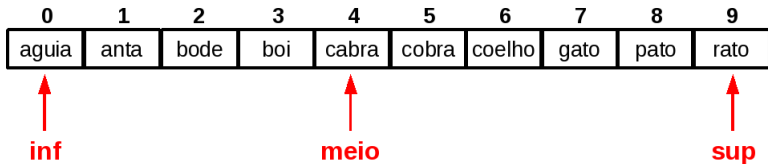
- Procurar o elemento: coelho (Passo 1—intervalo à direita)

0	1	2	3	4	5	6	7	8	9
aguia	anta	bode	boi	cabra	cobra	coelho	gato	pato	rato

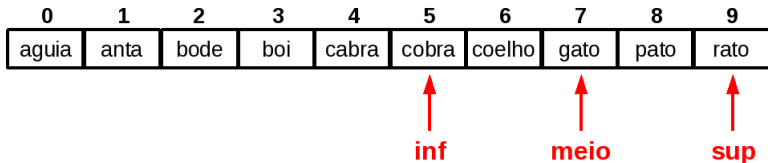
inf **meio** **sup**

Lista Estática Sequencial

Pesquisa Binária

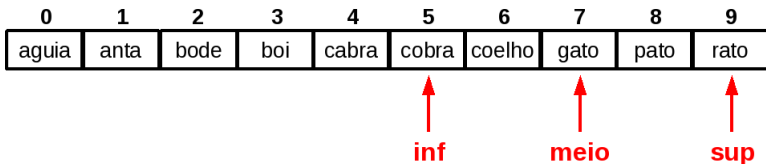


- Procurar o elemento: coelho (Passo 2—intervalo à esquerda)

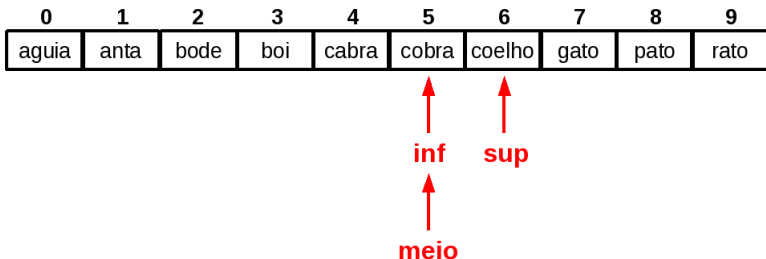


Lista Estática Sequencial

Pesquisa Binária



- Procurar o elemento: coelho (Passo 3—intervalo à esquerda)



Lista Estática Sequencial

Pesquisa Binária

- Procurar o elemento: coelho (Passo 4—intervalo à direita)

