Algoritmos e Estrutura de Dados - Listas

Eduardo Augusto Costa Trindade

⟨eduardo.trindade@ifmg.edu.br⟩

Sistemas de Informação

Instituto Federal de Ciência e Tecnologia de Minas Gerais - Campus São João Evangelista

INSTITUTO FEDERAL
MINAS GERAIS

Listas Lineares

- Uma das formas mais simples de interligar os elementos de um conjunto.
- Estrutura em que as operações **inserir**, **retirar** e **localizar** são definidas.
- Podem crescer ou diminuir de tamanho durante a execução de um programa, de acordo com a demanda.

INSTITUTO FEDERAL
MINAS GERAIS

Listas Lineares

- Duas listas podem ser concatenadas para formar uma lista única, ou uma pode ser partida em duas ou mais listas.
- Adequadas quando não é possível prever a demanda por memória.
  - Permite a manipulação de quantidades imprevisíveis de dados
- Aplicabilidade
  - São úteis em aplicações tais como manipulação simbólica, gerência de memória, simulação e compiladores.

INSTITUTO FEDERAL
MINAS GERAIS

Listas Lineares

- Sequência de zero ou mais itens $x_1, x_2, x_3 \cdots x_n$, na qual x_i é de um determinado tipo e n representa o tamanho da lista linear.
- Sua principal propriedade estrutural envolve as posições relativas dos itens em uma dimensão.
  - Assume-se que $n \geq 1$, x_1 é o primeiro item da lista e x_n é o último item da lista.
  - x_i procede x_{i+1} para $i = 1, 2, 3 \cdots n - 1$
  - x_i sucede x_{i-1} para $i = 2, 3 \cdots n$
  - O elemento x_i é dito estar na i-ésima posição da lista

INSTITUTO FEDERAL
MINAS GERAIS

TAD Listas Lineares

- O conjunto de operações a ser definido depende de cada aplicação.
- Um conjunto de operações necessário a uma maioria de aplicações consiste em:
  - Criar uma lista linear vazia.
  - Inserir um novo item imediatamente após o *i*-ésimo item.
  - Retirar o *i*-ésimo item.
  - Localizar o *i*-ésimo item para examinar e/ou alterar o conteúdo de seus componentes
  - Combinar duas ou mais listas lineares em uma lista única.

INSTITUTO FEDERAL
MINAS GERAIS

TAD Listas Lineares

- Um conjunto de operações necessário a uma maioria de aplicações consiste em:
  - Partir uma lista linear em duas ou mais listas.
  - Fazer uma cópia da lista linear.
  - Ordenar os itens da lista em ordem ascendente ou descendente, de acordo com alguns de seus componentes.
  - Pesquisar a ocorrência de um item com um valor particular em algum componente.

Implementações de Listas Lineares

- Várias estruturas de dados podem ser usadas para representar listas lineares, cada uma com vantagens e desvantagens particulares.

- As duas representações mais utilizadas são as implementações por meio de arranjos e de estruturas auto-referenciadas.

Implementações de Listas Lineares

- Exemplo de Conjunto de Operações:

Implementações de Listas Lineares

- Exemplo de Conjunto de Operações:
  - CriaLista(). Cria uma lista vazia.

Implementações de Listas Lineares

- Exemplo de Conjunto de Operações:
  - CriaLista(). Cria uma lista vazia.
  - Insere(x). Insere x após o último item da lista.

INSTITUTO FEDERAL
MINAS GERAIS

Implementações de Listas Lineares

- Exemplo de Conjunto de Operações:
  - CriaLista(). Cria uma lista vazia.
  - Insere(x). Insere x após o último item da lista.
  - Retira(x). Retorna o item x que está na posição i da lista, retirando-o da lista e deslocando os itens a partir da posição $i + 1$ para as posições anteriores.

INSTITUTO FEDERAL
MINAS GERAIS

Implementações de Listas Lineares

- Exemplo de Conjunto de Operações:
  - CriaLista(). Cria uma lista vazia.
  - Insere(x). Insere x após o último item da lista.
  - Retira(x). Retorna o item x que está na posição i da lista, retirando-o da lista e deslocando os itens a partir da posição $i + 1$ para as posições anteriores.
  - VerificaListaVazia(). Esta função retorna *true* se lista vazia; senão retorna *false*.

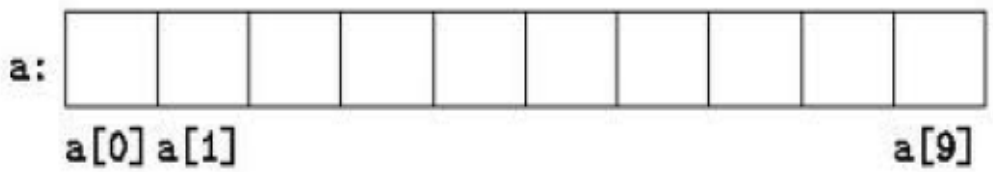
INSTITUTO FEDERAL
MINAS GERAIS

Implementações de Listas Lineares

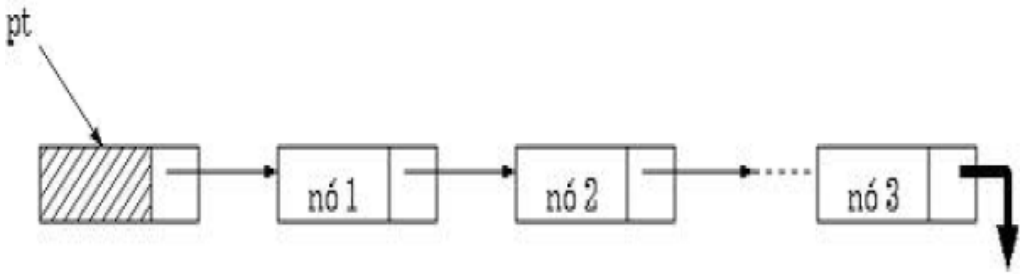
- Exemplo de Conjunto de Operações:
  - CriaLista(). Cria uma lista vazia.
  - Insere(x). Insere x após o último item da lista.
  - Retira(x). Retorna o item x que está na posição i da lista, retirando-o da lista e deslocando os itens a partir da posição $i + 1$ para as posições anteriores.
  - VerificaListaVazia(). Esta função retorna *true* se lista vazia; senão retorna *false*.
  - Imprime(). Imprime os itens da lista na ordem de ocorrência.

Implementações de Listas Lineares

- Várias estruturas de dados podem ser usadas para representar listas lineares, cada uma com vantagens e desvantagens particulares.
- As duas representações mais utilizadas são as implementações por meio de arranjos e de apontadores.
  - Arranjos



  - Lista Encadeada



Implementações de Listas por meio de Arranjos

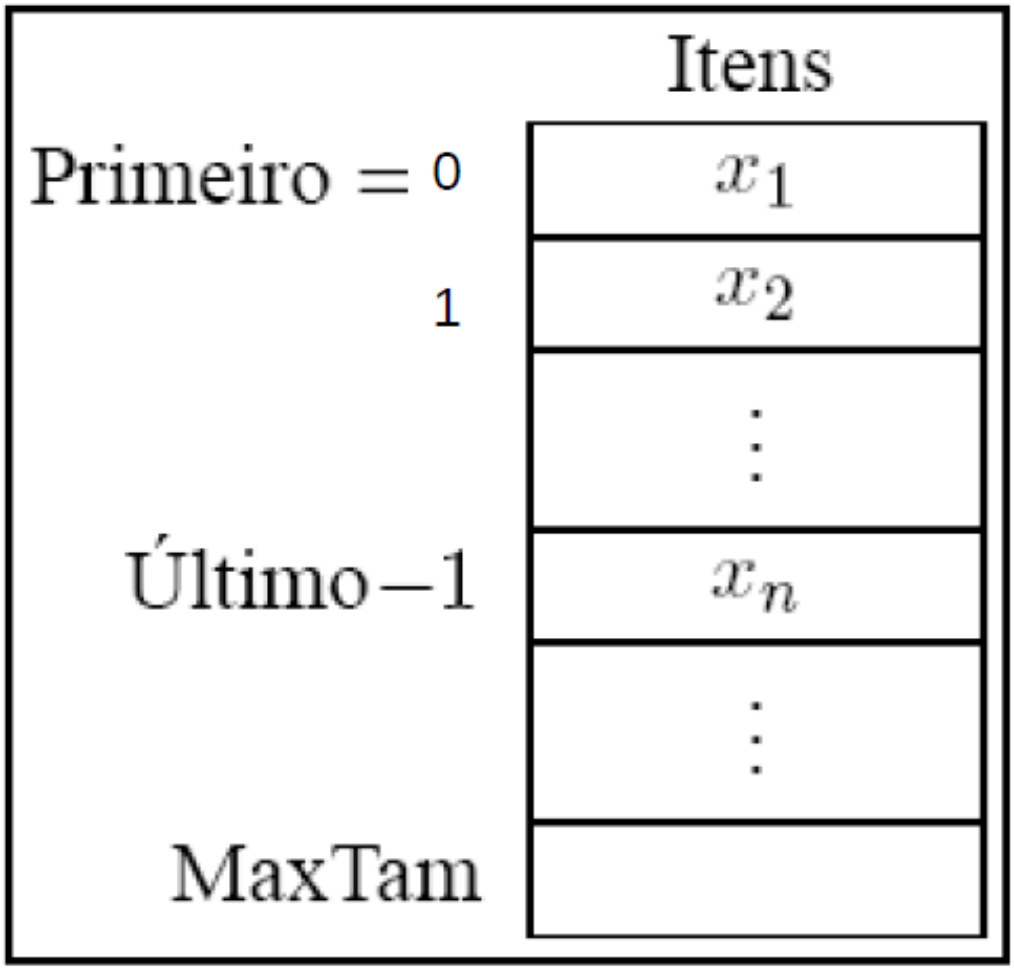
- Vantagens
  - economia de memória (os apontadores são implícitos nesta estrutura).
- Desvantagens
  - Custo para inserir ou retirar itens da lista, que pode causar um deslocamento de todos os itens, no pior caso.
  - Em aplicações em que não existe previsão sobre o crescimento da lista, a utilização de arranjos em linguagens como C/C++ pode ser problemática, porque o tamanho máximo da lista tem de ser definido em tempo de compilação.

Implementações de Listas por meio de Arranjos

- Os itens da lista são armazenados em posições contíguas de memória.
- A lista pode ser percorrida em qualquer direção.
- A inserção de um novo item pode ser realizada após o último item com custo constante.
- A inserção de um novo item no meio da lista requer um deslocamento de todos os itens localizados após o ponto de inserção.
- Retirar um item do início da lista requer deslocamento de itens para preencher o espaço deixado vazio.

INSTITUTO FEDERAL
MINAS GERAIS

Implementações de Listas por meio de Arranjos



Implementações de Listas por meio de Arranjos

- Conjunto de variáveis do mesmo tipo.
- Declaração $< tipo >< nome >[< tamanho >]$;
- Compilador aloca espaço de memória igual ao número de bytes do tipo vezes tamanho do vetor.
- *Bytes* alocados de maneira contígua na memória.
- Índice iniciando de 0 para acessar posições do arranjo.
- Limites do arranjo não são verificados pelo compilador na indexação.

INSTITUTO FEDERAL
MINAS GERAIS

Implementações de Listas por meio de Arranjos - Principais Funções

- void CriaListaVazia(TipoLista *lista);
- void VerificaListaVazia(TipoLista lista);
- void InsereLista(TipoLista *lista, TipoItem item);
- void RetiraLista(TipoLista *lista, TipoItem *item);
- void ImprimeLista(TipoLista lista);

Implementações de Listas por meio de Arranjos - Principais Funções

- void CriaListaVazia(TipoLista *lista);
- void VerificaListaVazia(TipoLista lista);
- void InsereLista(TipoLista *lista, TipoItem item);
- void RetiraLista(TipoLista *lista, TipoItem *item);
- void ImprimeLista(TipoLista lista);

* Pode-se utilizar **ponteiros** no lugar do retorno

INSTITUTO FEDERAL
MINAS GERAIS

Listas por meio de Arranjos - Prática

- Considerando as estruturas abaixo crie as principais funções para manusear uma lista.
- struct TipoItem{
  int Chave;
  /* outros componentes */
  }
- struct TipoLista{
  TipoItem item[maxTam];
  /* outros componentes */
  }

INSTITUTO FEDERAL
MINAS GERAIS

Listas por meio de Arranjos - Prática

- Adicione as funções abaixo para as informações criadas anteriormente.
- Concatenar listas.
- Intercalar (*Merge*).
- Copiar Lista.

Listas por meio de Arranjos - Prática

- Considerando as seguintes declarações de uma lista criada a partir da estrutura:

```
struct item {
        char nome[81];
        char matricula[8];
        char turma;
        float p1, p2, p3;
};
typedef struct lista TipoLista;
```

Listas por meio de Arranjos - Prática

- Implemente uma função que imprima o número de matrícula, o nome, a turma e a média de todos os alunos que pertencem a uma determinada turma. Os dados de cada aluno da turma selecionada devem ser impressos em uma única linha. Essa função deve obedecer o protótipo:

  void imprime_turma (Lista turmas, char turma)


Obrigado!