UDESC

ESTRUTURAS ESTÁTICAS – LISTAS

Conceito
Listas Sequenciais

Prof. Dr. Fabio Fernando Kobs

Listas – Definição e Características

Uma lista sequencial é um tipo abstrato de dados (ADT), e consiste de um conjunto de dados dispostos um depois do outro.

Um dos aspectos que diferem listas de estruturas de dados, como filas e pilhas, é a sua flexibilidade no processo de adição e remoção de itens. Novos itens podem ser adicionados tanto no início quanto no fim de suas extremidades, por isso, essa estrutura é conhecida como deque (do inglês *double ended queue*).

O mesmo acontece com itens existentes, que podem ser removidos de qualquer uma das extremidades. Por esse motivo, pode-se dizer que uma deque é uma estrutura "híbrida" que reúne as características de pilhas e filas em uma única estrutura de dados.

Listas - Características

A lista sequencial deve conter elementos de um mesmo tipo de dados, os quais estão dispostos em sequência lógica e também em sequência física.

A sua implementação é feita em forma de **vetor**, que pode estar **ordenado ou não** e ter elementos repetidos ou não.

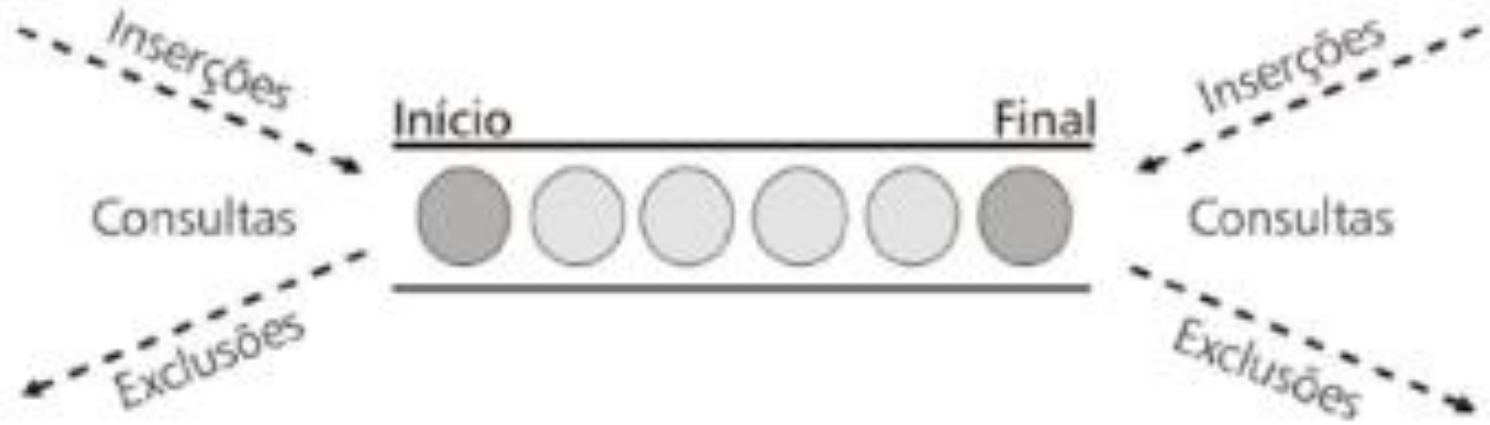
A alocação da memória é estática e a forma de armazenamento na memória é sequencial, ou seja, os elementos são armazenados em endereços de memória vizinhos.

Dentro de uma lista sequencial, os elementos ficam dispostos de forma consecutiva, sendo que nenhum deles pode ser nulo.

Quando um novo elemento é inserido, ou quando um dos elementos existentes é removido da lista, isso causa um deslocamento dos demais itens, a fim de torná-la organizada novamente, caso ela seja uma lista ordenada.

Listas - Características

Representação gráfica de uma deque, com suas possibilidades de inserções, consultas e exclusões.



Listas – Vantagens / Desvantagens

A utilização de uma lista estática sequencial traz como vantagens básicas:

- a possibilidade de acesso direto, por meio de índices, a qualquer um dos elementos da lista;
- um tempo de demora uniforme para efetivar o acesso a qualquer um dos elementos, pois isso vai depender apenas do índice.

Em contrapartida, a utilização desse tipo de lista também traz algumas desvantagens:

- movimentação, em alguns casos, de toda a lista para que um elemento seja inserido ou removido;
- o tamanho máximo deve ser predefinido e não poderá ser redefinido em tempo de execução da aplicação.

Listas - Funcionamento

Quando se trabalha com uma estrutura do tipo lista, sabe-se:

- qual é o seu primeiro elemento;
- que elemento vem depois de um determinado elemento;
- qual é o seu último elemento;
- quantos elementos existem na lista toda;
- como se faz para inserir um elemento na lista;
- como se faz para remover um elemento da lista.

O funcionamento de uma lista (inserção, remoção, atualização e consulta) se dá em qualquer ponto da lista.

Listas – Operações

Algumas das **operações** que podem ser executadas com listas sequenciais são:

- **Criação da lista**: a sua criação acontece quando se declara o vetor que vai conter os seus elementos.
- **Inicialização da lista**: envolve a preparação da lista para que sejam inseridos os elementos. Inicialmente, a lista possui 0 elementos.
- **Inserção de elemento**: se a lista estiver desordenada, consiste em inserir um elemento na primeira posição disponível do vetor e ajustar a quantidade de elementos caso a lista esteja cheia.
- **Percorrer a lista**: envolve mostrar os valores de todos os elementos da lista em qualquer direção.

Listas – Operações

- **Pesquisar na lista**: consiste na busca sequencial, de elemento por elemento, de um valor ao longo da lista. Ou seja, consiste em procurar um valor dentro da lista.
- **Remoção de elemento**: envolve o ajustamento da organização dos elementos da lista e também da quantidade. É importante lembrar que um elemento só pode ser removido da lista se ela não estiver vazia e se ele for encontrado.
- **Concatenação de listas**: é a união de duas listas em uma lista única.
- **Partição de listas**: consiste em repartir uma lista em duas ou mais listas.

Listas – Inserção

- A inserção de uma lista sequencial levará em conta que ela pode ou não estar ordenada, e pode ou não conter elementos repetidos. Nesse sentido, um elemento novo será inserido na posição equivalente ao valor da variável *n* e deverá ser feito um ajuste na quantidade de dados da lista, desde que ela já não esteja cheia.
- Para que um elemento novo seja inserido na lista, é preciso conhecer o vetor que a representa, o valor que precisa ser inserido, a quantidade atual de elementos da lista e qual o tamanho máximo da lista.
- Na função *main*, o usuário deverá informar um valor, o qual será enviado para uma função inserir, passando os parâmetros necessários.

Listas – Consulta

- A consulta em uma lista sequencial é uma operação que faz uma busca sequencial, elemento por elemento, a fim de encontrar um determinado valor, retornando à posição em que ele foi encontrado.
- Para que essa operação funcione, é necessário conhecer o vetor da lista, o valor que deve ser procurado e a quantidade atual de elementos da lista.
- Na função *main*, o usuário deverá informar um valor para ser consultado na lista, o qual será enviado para a função consultar, passando os parâmetros necessários.

Listas – Remoção

- A remoção de um determinado elemento de uma lista sequencial, que pode ou não estar ordenada, é uma operação que envolve o ajustamento do vetor e da sua quantidade, lembrando que a remoção só poderá ser feita se a lista não estiver vazia e se o elemento for encontrado (DEITEL; DEITEL, 2011).
- Para implementar essa função, é preciso conhecer o vetor da lista, o valor a ser removido da lista e a quantidade de elementos existentes na lista.
- Na função *main*, o usuário deverá informar um valor a ser removido e será feita a chamada da função remover, passando os parâmetros necessários.

Listas – Atualização

- A atualização de um elemento da lista envolve uma operação semelhante à consulta, pois é necessário varrer a lista para encontrar o elemento a ser alterado. Para que essa operação funcione, é necessário conhecer o vetor da lista, o valor que deve ser procurado, o novo valor do elemento e a quantidade de elementos atual da lista.
- Na função *main*, o usuário deverá informar um valor para ser consultado na lista e também um valor novo para esse elemento. Deverá ser feita uma chamada para a função atualizar, passando os parâmetros necessários.

Listas – Utilização

- As listas sequenciais são utilizadas, preferencialmente, quando é necessário manipular listas pequenas, quando se pode limitar o seu tamanho e quando a inserção e a remoção de um elemento puderem ser realizadas utilizando a última posição.
- As aplicações que manipulam listas, como a escalação de um time de futebol, os alunos de uma determinada disciplina, os funcionários de um setor, entre tantas outras que se encaixem nesses critérios, consistem em aplicações para listas sequenciais.

Ordenamento de uma Lista

Para certas aplicações será necessário manter os dados em ordem classificada na lista. Uma lista com essa característica é chamada de **lista ordenada**.

Em uma lista ordenada, os itens são organizados na ordem especificada (crescente ou decrescente).

A eliminação está geralmente limitada ao menor item (ou maior) na lista.

Ordenamento de uma Lista

Não há regras quanto à obrigatoriedade de inserção direta em uma lista ordenada.
Todavia, pode-se executar operações de ordenação antes de um processo específico, como por exemplo, listar os itens da lista de forma ordenada.

Listas – Python – Biblioteca *collections*

A biblioteca *collections* em Python automatiza a implementação desses métodos de manipulação.

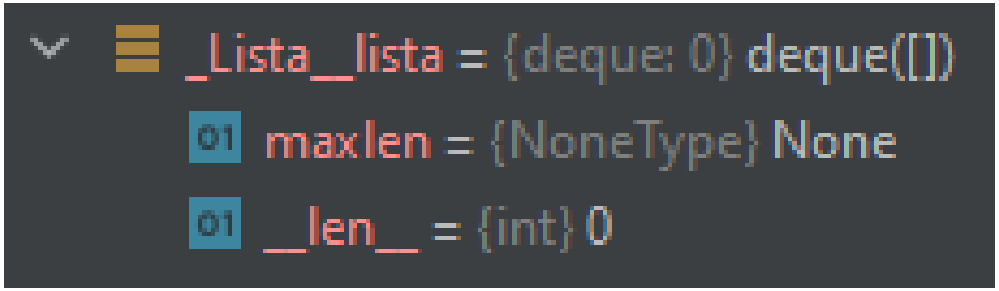
- *append*(x) — adiciona “x” ao lado direito da deque;
- *appendleft*(x) — adiciona “x” ao lado esquerdo da deque;
- *insert*(i, x) — insere o elemento “x” dentro da deque, na posição “i”;
- *pop*() — remove e retorna um elemento do lado direito da deque;
- *popleft*() — remove e retorna um elemento do lado esquerdo da deque;
- *remove*(*value*) — remove a primeira ocorrência do valor passado como parâmetro;
- *clear*() — deleta todos os elementos, deixando com tamanho igual a 0.

Listas – Python – Exemplo usando deque

```
import collections


class Lista:
    def __init__(self):
        self.__lista = collections.deque()
    def push_first(self):
        self.__lista.appendleft(input("Digite um nome para inserir: "))


if __name__ == '__main__':
    l = Lista()
    while True:
        op = int(input("\n1 - Inserir início\n2 - Sair\n\nOpcao: "))
        if op == 1:
            l.push_first()
        else:
            break
```



Exercícios - Lista por Vetor

Sugere-se que, a cada manipulação da lista, a mesma seja impressa em sua totalidade.

01) Construa um programa para incluir uma lista de nomes (inclusão no final da lista).

02) Altere o programa anterior, para permitir a inclusão de um nome no início da lista.

03) Altere o programa anterior, para permitir a inclusão de um nome no meio da lista, ou após uma "chave".

04) Altere o programa anterior, para permitir a exclusão do primeiro nome.

05) Altere o programa anterior, para permitir a exclusão do último nome.

06) Altere o programa anterior, para permitir a exclusão de um nome no meio da lista, na primeira ocorrência que encontrar.

07) Altere o programa anterior, para permitir a exclusão de todos os nomes da lista.

08) Refaça o exercício 07, mas agora, que permita manipular inteiros.

Exercícios - Lista por Vetor

Sugere-se que, a cada manipulação da lista, a mesma seja impressa em sua totalidade.

09) Construa um programa para incluir uma **lista ordenada** (em ordem crescente) de nomes.

10) Altere o programa anterior, para permitir a exclusão de um nome na lista (na primeira ocorrência que encontrar).

11) Altere o programa anterior, para permitir a exclusão de todos os nomes da lista.

12) Refaça o exercício 11, mas agora, que permita manipular inteiros.

Exercício 13

Construa um programa que, a partir da lista L desordenada, tenha uma função que cria uma nova lista K ordenada, com os mesmos dados da lista L.
A função deve remover os elementos da lista L, e inserir na lista K, que dessa forma torna-se uma lista ordenada (em ordem crescente).

Observações:

- No final do processo, a lista L estará vazia e a lista K conterá os dados anteriormente alocados para a lista L.

Exercício 14

Construa um programa com uma função para concatenar duas listas L1 e L2 em uma única lista ordenada. Depois, inclua uma função para remover os elementos repetidos da lista ordenada. Imprimir as listas.

REFERÊNCIAS

- CURY, Thiago E.; BARRETO, Jeanine dos S.; SARAIVA, Maurício de O.; et al. Estrutura de Dados. Grupo A, 2018. E-book. ISBN 9788595024328. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595024328/. Acesso em: 25 mar. 2024.
- DEITEL, P.; DEITEL, H. C: como programar. 6. ed. São Paulo: Pearson Education, 2011. 818 p.
- PINTO, Rafael A.; PRESTES, Lucas P.; SERPA, Matheus da S.; et al. Estrutura de dados. Grupo A, 2020. E-book. ISBN 9786581492953. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786581492953/. Acesso em: 25 mar. 2024.