



Listas Encadeadas Ordenadas (ordered linked lists)

Objetivos

1. Compreender a diferença de implementação de listas encadeadas ordenadas e não ordenadas.
2. Ser capaz de implementar listas encadeadas ordenadas
3. Ser capaz de avaliar o desempenho de listas encadas ordenadas

Conteúdo

1. Métodos search e add
2. Análise do desempenho das listas ordenadas

Listas Encadeadas Ordenadas

- Há uma relação de ordem entre os elementos da lista



Funções: * ListaOrdenada() cria uma nova ListaOrdenada vazia * add(item) adiciona um novo item na lista garantindo que a ordem é preservada * remove(item) remove um elemento da lista * search(item) busca um item na lista. retorna True ou False * isEmpty() verifica se a lista está vazia * size() retorna o número de itens na lista * index(item) retorna a posição de um item na lista * pop() remove e retorna o último elemento da lista * pop(pos) remove e retorna o elemento na posição pos]

Lista ordenada

- Usa as mesmas técnicas de lista não ordenada
- Mas é possível se valer da ordenação para deixar as funções mais rápidas

Lista ordenada (search)

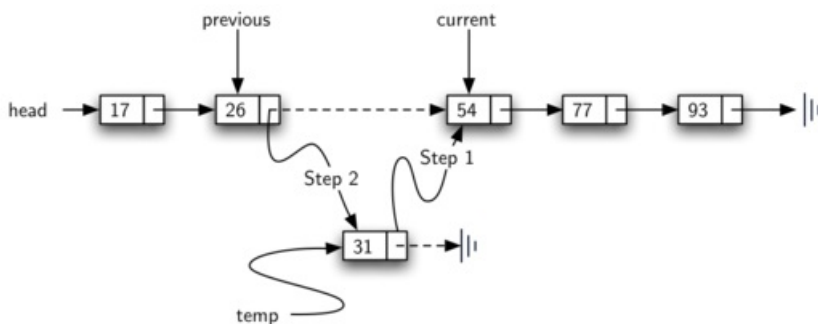
```

def search(self,item):
    atual = self.head #atual == temp
    encontrou = False
    parar = False
    while atual != None and not encontrou and not parar:
        if atual.getData() == item:
            encontrou = True
        else:
            if atual.getData() > item:
                parar = True
            else:
                atual = atual.getNext()

    return encontrou

```

Lista ordenada (add)



Lista ordenada (add) [2]

```

def add(self,item):
    atual = self.head
    anterior = None
    parar = False
    while atual != None and not parar:
        if atual.getData() > item:
            parar = True
        else:
            anterior = atual
            atual = atual.getNext()

    temp = Noh(item)
    if anterior == None:
        temp.setNext(self.head)
        self.head = temp
    else:
        temp.setNext(atual)
        anterior.setNext(temp)

```

Análise do desempenho da implementação da lista encadeada

Operação	Tempo de Execução
add	$O(n)$
remove	$O(n)$
search	$O(n)$
is_empty()	$O(1)$
size()	$O(N)$ ou $O(1)$ na versão melhorada

Visualize em <https://visualgo.net/pt/list>

Para estudar

- Filas
 - Seções 3.22 a 3.23 do livro [5] https://panda.ime.usp.br/pythonds/static/pythonds_pt/index.html
 - Seção 2.1 do livro [4]
 - Capítulo 17 (Litas encadeadas) do livro [2] (texto bastante resumido!)
<https://chevitarese.files.wordpress.com/2009/09/aprendacomputaocompython3k.pdf>

Referências

1. Tradução do livro *How to Think Like a Computer Scientist: Interactive Version*, de Brad Miller e David Ranum. link: <https://panda.ime.usp.br/pensepy/static/pensepy/index.html>
2. Allen Downey, Jeff Elkner and Chris Meyers. *Aprenda Computação com Python 3.0*. link: <https://chevitarese.files.wordpress.com/2009/09/aprendacomputaocompython3k.pdf>
3. SANTOS, A. C. *Algoritmo e Estrutura de Dados I*. 2014. Disponível em <http://educapes.capes.gov.br/handle/capes/176522>
4. SANTOS, A. C. *Algoritmo e Estrutura de Dados II*. 2014. Disponível em 2014. Disponível em <https://educapes.capes.gov.br/handle/capes/176557>
5. Tradução do livro [5] *Problem Solving with Algorithms and Data Structures using Python* de Brad Miller and David Ranum. link: https://panda.ime.usp.br/pythonds/static/pythonds_pt/index.html
6. Brad Miller and David Ranum. *Problem Solving with Algorithms and Data Structures using Python* link: https://panda.ime.usp.br/pythonds/static/pythonds_pt/index.html
7. Caelum. Algoritmos e Estruturas Dados em Java. Disponível em <https://www.caelum.com.br/download/caelum-algoritmos-estruturas-dados-java-cs14.pdf>

That's all Folks