

EDA 2021/2022**Trabalho 2 – trabalho de grupo intermédio****Exploração do TAD Lista com implementação de listas ligadas**

Deve trabalhar nas respostas às questões colocadas e submeter o resultado do seu trabalho até à **data limite de entrega: 18 de abril de 2022** (23h59m).

Submissão: fazer o carregamento (upload) do ficheiro ZIP de resposta no e-Learning Blackboard, na página do trabalho.

Introdução

O objetivo deste trabalho é explorar as representações do TAD Lista em Python e os algoritmos de pesquisa e de ordenamento. Cada grupo de trabalho deve fazer **duas** implementações diferentes do TAD Lista usando representações dinâmicas e utilizar um algoritmo de pesquisa e outro de ordenação para implementar as operações do interface TAD Lista que requerem pesquisa e ordenação. Estas implementações têm de respeitar os requisitos que são detalhados em baixo. A última parte do trabalho consiste em realizar uma análise temporal empírica dos algoritmos implementados e realizar uma análise espacial assintótica.

Este trabalho de grupo intermédio, tal como o trabalho de grupo final é para ser realizado por grupos de 3 ou 4 elementos. Cada grupo de trabalho deve ser registado de acordo com as instruções que serão brevemente publicadas no Blackboard.

Fase 1 - TAD Lista: implementações

Criar duas implementações para o TAD Lista, ambas usando uma representação dinâmica e escolhendo uma das opções seguintes:

- a) simplesmente ligada e circular
- b) simplesmente ligada e duplamente ligada
- c) duplamente ligada e circular (opção com valoração superior).

Pretende-se uma implementação do TAD Lista, usando uma lista ligada que obedeça à seguinte interface:

- 1) comprimento ou tamanho: quantos itens estão na lista – len()
- 2) indicar se a lista está vazia – vazia()
- 3) esvaziar a lista – limpar()
- 4) consultar (devolver) elemento na posição p – ver(p)
- 5) inserir um dado item na lista - ins(item)
- 6) remover um determinado item da lista - rem(item)
- 7) mostrar o conteúdo completo da lista – mostrar()
- 8) indicar se existe na lista um item com um dado valor – existe(item)
- 9) ordenar a lista – ordenar().

Explicar e justificar as opções de representação escolhidas em cada uma das implementações, bem como opções para os métodos de pesquisa ou ordenamento (por exemplo, métodos auxiliares criados para facilitar algum destes). **A boa modularidade do código e boa documentação serão valorizadas.**

Nota 1: O método 8 deve devolver 0 (zero) se o item procurado não existir ou i se o item foi encontrado na i-ésima posição e deve usar uma pesquisa sequencial. O método 9 deve receber um parâmetro que indica qual a ordenação a usar (por exemplo: b indica bubblesort e q indica quicksort), usando uma inicialização por omissão na declaração do método para o ordenamento considerado mais eficiente após a fase 2.

Nota 2: As operações da interface TAD Lista que requerem pesquisa ou ordenação devem ser implementadas usando os algoritmos de pesquisa e ordenação escolhidos pelo grupo de entre uma das opções descritas na subsecção seguinte (1.1 Algoritmos de pesquisa e ordenação).

1.1 Algoritmos de pesquisa e ordenação

Implementar o algoritmo de **pesquisa binária** e **dois** algoritmos de **ordenamento** (um da classe (a) e outro da classe (b)) à escolha entre cada uma das seguintes combinações:

- a. Ordenação por Mergesort ou por Quicksort;
- b. Ordenação por Bubblesort ou por Insertionsort.

Nota 3: Só deve escolher dois algoritmos de ordenação e utilizar nas duas implementações.

Fase 2 - Análise da complexidade de algoritmos

Esta fase só será aceite caso tenham realizado, pelo menos, uma implementação de uma das representações com os métodos 1 a 9.

1. Tal como no exercício 4 da última aula laboratorial, deve realizar uma análise temporal empírica aos algoritmos escolhidos e às duas implementações. Deve fazer testes ao comportamento (desempenho) das suas implementações, representar graficamente e descrever que conclusões se podem tirar sobre o crescimento da complexidade temporal dos algoritmos, para cada uma das representações do TAD Lista implementadas e em confronto com a complexidade conhecida de cada um dos métodos.
2. Realizar também uma análise espacial assintótica e explicar as diferenças tendo por base as duas implementações do TAD Lista.

Submissão do trabalho incluindo relatório

O trabalho deve ser distribuído entre os 3 ou 4 elementos do grupo, devendo estar equitativamente distribuído. Apesar de **todos deverem trabalhar em conjunto para a conclusão deste trabalho**, cada elemento do grupo ficará diretamente **responsável** por uma parte do desenvolvimento do trabalho (por exemplo: a aluna A é responsável pela implementação dos métodos 1 a 8 da fase 1, o aluno B é responsável pela implementação da pesquisa binária e métodos de ordenamento e a aluna C pela implementação e análise na fase 2). Nos grupos com 4 elementos, o 4.º elemento deve ficar responsável pela organização e escrita do relatório do projeto. No final do relatório, devem **indicar a responsabilidade de cada elemento do grupo**.

Cada grupo deve submeter o resultado do seu trabalho até à data limite de entrega: 18 de abril de 2022 (23h59m).

Para a submissão, devem entregar um zip com:

- um ficheiro Jupyter notebook, que deve incluir o código Python desenvolvido (ou, pelo menos, as partes importantes) e as justificações e discussões pedidas.
- um ficheiro com o código Python desenvolvido, em formato (.py) pronto a ser utilizado/testado.

A partilha de código em trabalhos de grupos diferentes será devidamente penalizada. Serão usados os meios habituais de verificação de plágio. O código entregue deve ser integralmente da autoria dos membros do grupo.

Notas finais: É natural que sintam dificuldades na resolução completa deste trabalho. Tentem responder o melhor que conseguirem às questões colocadas, mas, se não conseguirem, não copiem de alguém ou de algum lado sem perceber o que cada linha de código faz, e coloquem esta dificuldade a algum dos docentes ou conversem com outra(o) colega. Neste último caso, cada grupo deve indicar no trabalho, em comentário inicial, que conversou sobre o assunto X com a(o) colega Y.

Lembre-se que esta unidade curricular está a preparar para criar competências que serão necessárias até ao final da Licenciatura e muito para além desta. Não vai ter sempre a possibilidade de apresentar algo feito por outros!