Trabalho 1 - versão 1 – publicada em 17/Março/2023

Carregamento e ordenação de *Big Data*

Este primeiro trabalho tem como objetivo encontrar o valor máximo e mínimo de uma lista de valores com 100 000 observações. A resolução deste problema deve ter em vista a complexidade temporal de algoritmos, de modo a resolver o problema da forma mais eficiente possível. Os valores relativos às observações serão fornecidos numa base de dados usando o formato CSV (*Common-separated value*).

O trabalho é dividido nas seguintes fases:

a) Leitura do CSV

Tendo em conta que a base de dados apresenta um número muito elevado de observações, não é eficiente lê-la na integra. Para resolver esta questão a base de dados deve ser lida em partições de **n** observações. Para concretizar este objetivo os alunos devem pesquisar funções do package pandas que permitam ler a base de dados, fornecida através de um ficheiro CSV, por partes. Informação sobre o package pandas (Python Data Analysis Library) pode ser encontrado em https://pandas.pydata.org/

Deve ser desenvolvida a **função partitions(data, n)** que recebe a base de dados (**data**) e o número de observações por partição (**n**). O output da função é um iterador sobre as várias partições do CSV.

b) Sorting

Desenvolver a função **sorting(list, ascending)** que recebe uma lista de valores não ordenados e devolve a lista ordenada, o número de iterações que o algoritmo levou a concluir a ordenação, o mínimo e o máximo da lista. O parâmetro **ascending** permite ordenar a lista por ordem crescente ou decrescente. Desenvolver a função **sorting** usando um dos algoritmos de ordenação explicados em aula.

c) Função execução

Deve ser desenvolvida a função **execute(data, n, ascending)**. Esta função recebe a base de dados, o critério de ordenação e o número de observações por partição. Esta função deve iterar sobre as várias partições do CSV, sendo que, para cada uma são ordenados os seus valores numéricos. O output da função é uma tabela com os conteúdos indicados a seguir. Para cada partição do CSV devem ser guardadas numa tabela (**pandas.DataFrame**) as seguintes informações:

- O tempo de execução da iteração
- O valor máximo
- O valor mínimo
- Número de iterações que o sorting executa.

d) Experiências

Uma vez desenvolvidas as funções definidas em **a), b)** e **c)** é possível efetuar várias experiências experimentando vários valores de **n** (número de observações por partição). Para cada experiência deve ser guardado o tempo total que o código demorou a ser executado.

Com esta informação, desenvolver um gráfico utilizando o módulo **matplotlib** para ver graficamente qual o valor de **n** que minimiza o tempo de execução do código.

Aspetos de qualidade do relatório e do código Python

- Respeitar as normas PEP 8 para o desenvolvimento de código Python
- Elementos que devem estar contidos no relatório:
    - Identificação dos elementos do grupo
    - Data do relatório
    - Código Python desenvolvido, conforme o pedido no enunciado
    - Algoritmo de ordenação escolhido
    - Experiências realizadas
        - explicar as hipóteses e decisões que foram tomando
        - resultados obtidos, incluindo os gráficos
        - análise e discussão.

Submissão do trabalho incluindo relatório

O trabalho deve ser distribuído entre os 2 elementos do grupo, devendo estar equitativamente distribuído. Cada grupo deve submeter o resultado do seu trabalho até à data e hora limite de entrega: 17 de abril de 2023, 23h59m. A submissão do trabalho deve ser feita no Moodle na Actividade do Trabalho 1, fazendo o carregamento (upload) do ficheiro ZIP com o conteúdo descrito de seguida.

Para a submissão, devem entregar um zip com:

- o relatório num formato que vos pareça adequado, que deve incluir o código Python desenvolvido (ou, pelo menos, as partes importantes) e os restantes elementos pedidos acima;
- um ficheiro com o código Python desenvolvido, em formato (.py) ou no formato (.ipynb) 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.