

Análise de Complexidade de Algoritmos

Os exercícios seguintes devem ser desenvolvidos nos packages `pt.ipleiria.estg.dei.aed.analisecomplexidade.algoritmos` e `pt.ipleiria.estg.dei.aed.analisecomplexidade.utilizacao`.

1. Determinar o valor máximo de uma sequência

Tomando como base uma sequência de inteiros, pretende-se **encontrar o valor máximo** nela presente.

- Implemente um algoritmo **iterativo** que produza o efeito supracitado;
- Execute a função para sequências de 5 e 10 inteiros gerados aleatoriamente de -50 a 50 e apresente a estatística da execução (número de comparações e tempo de execução) do algoritmo para os valores indicados;
- Apresente uma janela de gráficos para a execução do algoritmo que demonstre a evolução do número de comparações e do tempo de execução para sequências de tamanho 10 a 500 com incrementos de 10;
- Simule a execução do algoritmo no *JGrasp*;
- Estime o número de comparações efetuadas para uma sequência de tamanho 15 e **justifique** calculando o grau de complexidade do algoritmo.

2. Obter o par de pontos mais próximos

Tomando como base uma sequência de pontos cartesianos em \mathbb{R}^2 , pretende-se **obter o par de pontos mais próximos entre si**.

- Implemente um algoritmo **iterativo** que resolva o problema apresentado;
- Execute a função para sequências de 5, 10 e 15 pontos gerados aleatoriamente e apresente a estatística da execução (número de comparações e tempo de execução) do algoritmo para os valores indicados;

- c) Apresente uma janela de gráficos que demonstre a evolução do número de comparações e do tempo de execução do algoritmo para sequências de tamanho 3 a 20.
- d) Simule a execução do algoritmo no *JGrasp*;
- e) Estime o número de comparações feitas para uma sequência de 30 pontos e **justifique** calculando o grau de complexidade do algoritmo;

3. Torres de Hanói

Considere o algoritmo implementado na ficha prática 1.

- a) Efetue uma simulação no *JGrasp* apresentando as três torres com os respectivos discos. Para tal, considere que o tamanho de cada disco é representado por um número natural a iniciar em 1. Cada torre deve ter uma base com o triplo do tamanho do maior disco;

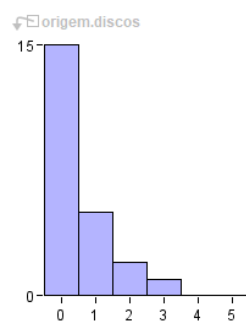


Figura 1. Representação da Torre Origem com base na posição 0, num passo intermédio, para 5 discos

- b) Usando a notação **Big-O**, estime o crescimento do tempo de execução com o crescimento do tamanho do problema;
- c) Comprove a estimativa com base na execução da implementação efetuada na ficha prática anterior.