

O presente projeto envolve a criação de uma biblioteca de classes respetivos métodos e testes que permitam gerir a informação relativa a vários países tais como, população, continente, capital e respetivas coordenadas (latitude, longitude) em graus decimais. A informação encontra-se nos ficheiros de texto:

`países.txt`: nome, continente, população, capital, latitude, longitude

`fronteiras.txt`: pais1, pais2

Usando a classe **árvore binária de pesquisa (BST)** apresentada nas aulas, desenvolva da forma **mais eficiente possível** as seguintes funcionalidades:

1. Com recurso à estrutura árvore binária de pesquisa (BST):
 - a. Contruir uma árvore binária de pesquisa que contenha a informação relativa a cada país: nome, continente, população, capital, latitude, longitude, lista de países fronteira e número total de fronteiras. Forneça um método que, dado um nome de país, devolva a lista das suas fronteiras com outros países.
 - b. Fornecer um método capaz de devolver uma lista ordenada dos países pertencentes a um determinado continente. Efectuar ordenação decrescente por número de fronteiras e crescente por valor de população (i.e.: maior número de fronteiras e menor população).
2. A estrutura kd-tree é uma modificação da árvore binária de pesquisa (BST) que permite o processamento eficiente de chaves de pesquisa multidimensionais. Pretende-se com a utilização desta estrutura, e considerando duas dimensões ($k=2$), efectuar pesquisas de países por coordenadas (latitude, longitude) no planisfério da superfície da terra onde estão representadas as capitais de países numa 2d-tree.

A 2d-tree difere da BST na forma como toma as decisões de ramificação. Enquanto que na BST apenas é utilizada uma chave para decidir a ramificação (utilização de apenas um discriminador), na 2d-tree são utilizadas duas chaves. Isto é, o discriminador é escolhido de acordo com o nível correspondente na árvore (discriminador X para níveis pares e discriminador Y para níveis ímpares). O resultado é a divisão do plano em cortes verticais para valores de X e cortes horizontais para valores de Y como é ilustrado na Figura 1.

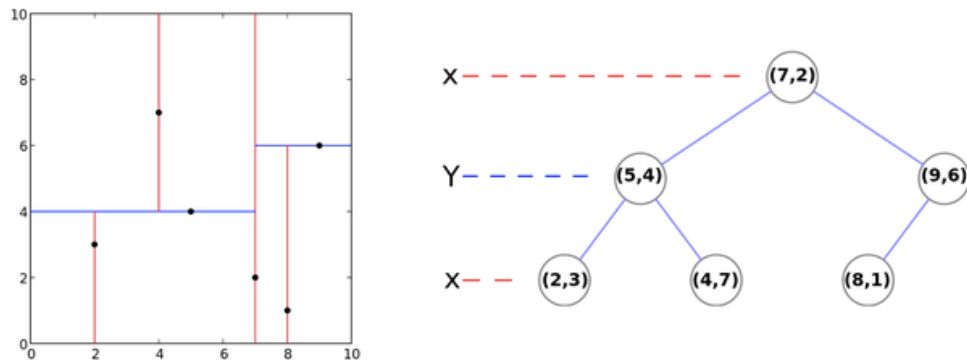


Figura 1 – Exemplo de uma kd-tree de duas dimensões ($k=2$). (a) decomposição por regiões contendo 6 coordenadas (x, y) . (b) árvore 2d-tree correspondente. De notar que nos níveis pares é utilizado o discriminador x e nos ímpares o y .

Com recurso à estrutura 2d-tree (kd-tree de $k=2$) e utilizando a forma mais eficiente de inserção e pesquisa:

- Construir uma 2d-tree** que contenha a informação relativa a cada país: nome, continente, população, capital, latitude, longitude, lista de países fronteira e número total de fronteiras. Considere x como latitude e y longitude.
- Pesquisa exata:** utilizando a 2d-tree da alínea a), fornecer um método que devolva o país cuja capital está situada nas coordenadas (latitude, longitude).
- Pesquisa vizinho mais próximo:** utilizando a 2d-tree¹ da alínea a), fornecer um método que encontre o país cuja capital está mais próxima das coordenadas (latitude, longitude).
- Pesquisa por área geográfica:** Utilizando a 2d-tree da alínea a), fornecer um método que devolva a lista de todos os países cuja capital está contida numa área dada por um rectângulo de coordenadas (latitude1, longitude1) e (latitude2, longitude2). A Figura 2 ilustra um exemplo de pesquisa por área rectangular.

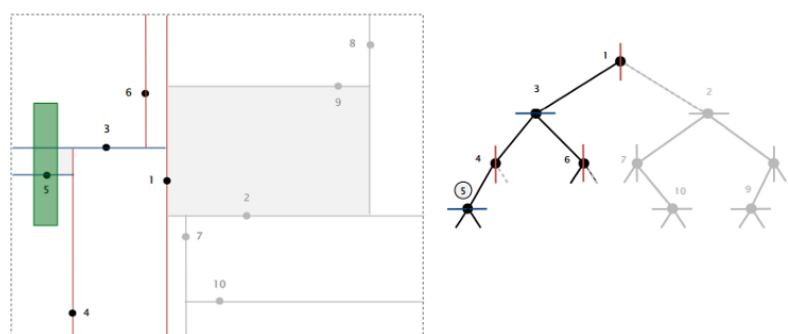


Figura 2 – Exemplo de pesquisa numa 2d-tree por área geográfica rectangular.

¹ <https://www.coursera.org/lecture/ml-clustering-and-retrieval/nn-search-with-kd-trees-6eTzw>

Normas

- A avaliação do trabalho será feita principalmente em função das classes propostas, nomeadamente em termos da sua conformidade com o Paradigma da Programação por Objetos e **eficiência** das estruturas de dados usadas e funcionalidades solicitadas.
- O trabalho deverá ser realizado em **grupos de dois alunos**. Os grupos têm de ser formados e enviados por *email* ao docente das aulas PL.
- O projeto tem de ser desenvolvido em Java e todas as funcionalidades testadas através de testes unitários e usando os ficheiros de teste disponibilizados.
- É obrigatório o uso da ferramenta de **controle de versões Git**.
- O relatório deverá ser elaborado para cada uma das partes do trabalho e deve servir de ferramenta de avaliação posterior à apresentação. Nele devem apresentar o digrama de classes, algoritmos dos métodos, análise de complexidade de todas as funcionalidades implementadas, melhoramentos possíveis.
- O trabalho deve ser submetido no Moodle até às **24 horas do dia 1 de Dezembro**. A partir desta data a nota do trabalho será penalizada **10% por cada dia de atraso** e não se aceitam trabalhos **após dois dias** das datas indicadas.
- Na semana seguinte à data de entrega o professor das aulas práticas fará a cada grupo de trabalho uma avaliação qualitativa do projeto submetido.
- A apresentação/avaliação do trabalho final será individual e realiza-se na 12ª semana de aulas, em datas a fixar com o professor das aulas práticas.