

K pontos vizinhos mais próximos

Um problema muito comum nas áreas de matemática e computação é o chamado *kNN* (*k Nearest Neighbor* – em português, k vizinhos mais próximos). Em sua forma mais simples, pode ser enunciado com o seguinte parágrafo:

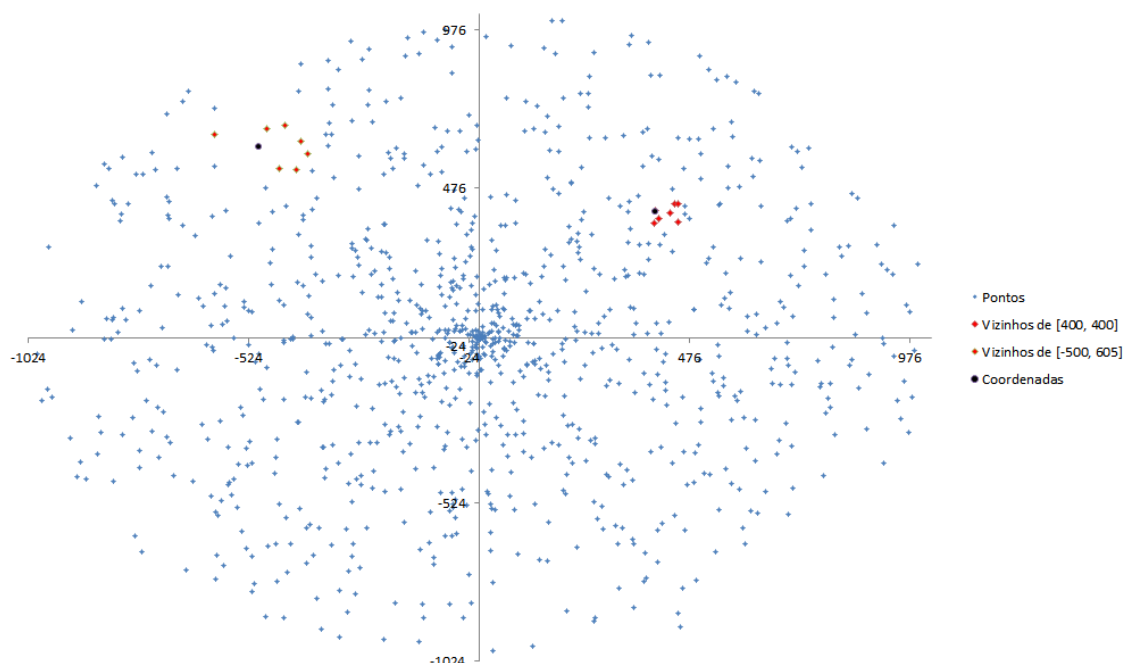
“Dado um conjunto de pontos P , num plano cartesiano de dimensão D , e uma coordenada C de mesma dimensão, encontrar os k pontos pertencentes a P mais próximos a C .”

Esse tipo de problema é muito relacionado, por exemplo, à busca de pontos de turismo próximos a uma coordenada qualquer num mapa (vide Google Maps). Também está relacionado a algoritmos de Data Mining em que, por exemplo, deseja-se saber quais itens/elementos um determinado perfil de usuário vai gostar (buscas personalizadas). É também relacionado a cálculo de movimentos e aceleração de planetas e estrelas (uma vez que os astros mais próximos são aqueles que exercem maior força gravitacional).

Um exemplo rápido pode ser dado pela imagem abaixo. Nela é mostrado um conjunto de 1024 pontos (de cor azul), e são destacados em vermelho os 7 pontos mais próximos às coordenadas [400, 400] e [-500, 605] (coordenadas destacadas em preto).

Objetivo

O objetivo desse projeto é implementar um sistema que tenha como entrada um arquivo com N pontos, e através de uma coordenada fornecida pelo usuário, faça a busca pelos k vizinhos mais próximos à essa coordenada, em seguida seu programa apresenta na tela os k pontos vizinhos mais próximos da coordenada informada e as respectivas distâncias, os k vizinhos deverão estar organizados em ordem crescente de distâncias. Considere que a coordenada fornecida pelo usuário e a quantidade de k vizinhos são lidos a partir do teclado, para este trabalho a dimensão $D = 2$ (ou seja, pontos com coordenadas num plano x e y).



Considere a métrica euclidiana para o cálculo da distância entre dois pontos num plano cartesiano, que é dada pela expressão abaixo:

$$dist(P_1, P_2) = \sqrt{(P_1.x - P_2.x)^2 + (P_1.y - P_2.y)^2}$$

Abaixo é apresentado o exemplo de um arquivo texto de entrada, na primeira linha temos um inteiro representando a quantidade de pontos que serão lidos, nas linhas seguintes dois valores x,y representando os N pontos de entrada.

```
100
10 15
18 40
-10 54
56 10
. . .
```

Restrições do projeto

1. Para representar os pontos do sistema você deve implementar uma classe **Ponto** e as operações referente a essa classe, por exemplo, o método que calcula a distância entre dois pontos. **Não** é permitida a utilização da classe **Point**, **Point2D** ou **Point3D** do Java.
2. Para armazenar os pontos do seu sistema você deverá usar **vetor de objetos** não é permitido usar classes prontas do Java para isso, como **ArrayList**. Os **k** pontos mais próximos também deverão estar armazenados em um **vetor de objetos**.
3. O programa deve estar bem documentado e implementado na linguagem Java, aplicando os conhecimentos sobre programação orientado a objetos e tipo abstrato de dados vistos nesse semestre.
4. Este trabalho pode ser desenvolvido em grupos de até **2 alunos**. Como este trabalho pode ser feito em grupo, evidentemente você pode “discutir” o problema dado com outros grupos, inclusive as “dicas” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa. Ou seja, qualquer tentativa de fraude será punida com a **nota zero**.
5. A entrega consistirá em uma apresentação do projeto ao professor responsável pela disciplina e será a avaliado de acordo com os seguintes critérios:
 - a) Funcionamento do programa;
 - b) O quão fiel é o programa quanto à descrição do enunciado;
 - c) Clareza na nomenclatura de variáveis, métodos e classes;
 - d) E principalmente os conceitos vistos de tipo abstrato de dados e programação orientada a objetos durante as aulas da disciplina;