Algoritmos e Estruturas de Dados LC 2015/2

Exercício Programa 2: Trem-bala de Bangladânia

Rodrigo de Souza

Entrega: 19/10/2015

A grande nação de Bangladânia pretende, como parte de grandes empreendimentos previstos para o campeonato mundial de truco que sediará, construir um trem-bala ligando duas de suas maiores cidades. Após anos de batalhas políticas, re-engenharias financeiras e imbróglios judiciais, os políticos de lá compreenderam que um trem-bala só seria viável se ligasse o pair mais próximo de suas cidades, e olhe lá.

A empresa que mais vem agradando no processo licitatório afirma que esse problema é comum nos (muitos) países onde já implantou linhas de trem-bala, por isso possui um sistema automatizado que, a partir do mapa das cidades de um país, encontra o par mais próximo de cidades (em termos de distância euclidiana no plano, sem desvios).

Mas a empresa, que estava mentindo só para ganhar a licitação, não tem sistema algum. Então você foi contratado às pressas para construir esse sistema. Além disso, para que sua empresa continue com chances de ganhar a licitação de grandes conglomerados industriais também interessados no projeto, sua solução deve ter um mínimo de sofisticação.

1 Entrada

A entrada deve ser lida de um arquivo `trembala.dat` e consiste de diversos casos de teste; cada caso de teste apresenta o mapa de um país, representado pelas coordenadas de suas cidades no plano cartesiano. Cada caso de teste ocupa três linhas: a primeira linha é o número n de cidades no mapa, ou seja, as cidades são numeradas como $1, 2, \ldots, n$. A segunda linha apresenta as abscissas e a terceira as ordenadas das cidades (separadas por espaços), ou seja, as coordenadas das cidades são $(x[1], y[1]), (x[2], y[2]), \ldots, (x[n], y[n])$, onde x e y são os vetores apresentados na segunda e terceira linhas respectivamente. Pode supor que as coordenadas são números inteiros. O fim da entrada é indicado por $n = 0$.

Para cada caso de teste, seu programa deve imprimir (na tela mesmo) um par de índices, indicando um par mais próximo de cidades no mapa, e a distância entre elas. A saída não deve conter nenhuma formatação além desses três números, separados por espaços.

Sua solução, para que atinja a sofisticação necessária, deve implementar uma estratégia de divisão-e-conquista que consiste em:

- Ordenar previamente as abscissas das cidades (isso só é feito uma vez; é um pré-processamento) usando o algoritmo de ordenação por intercalação que estudamos;
- Dividir o conjunto de pontos em dois subconjuntos de mesmo tamanho separados por uma linha vertical, usando para isso a ordenação produzida no pré-processamento;
- Resolver recursivamente o problema em cada um dos sub-conjuntos de pontos, e juntar as duas soluções para construir a solução do conjunto original.

Você pode encontrar uma descrição sucinta mas bastante completa dessa solução em

`https://www.ime.usp.br/~cris/aulas/10_1_6711/slides/aula2pmp.pdf`

Você não é obrigado a implementar a solução que resulta em complexidade $\mathcal{O}(n \lg n)$ (a parte de ordenar o eixo y); mas se o fizer e funcionar, terá um bônus de 1 ponto na nota.

2 Instruções gerais

- Seu programa deve ser feito em C, Java ou Python.
- Não tente embelezar a saída do seu programa com mensagens, formatações, etc. Seu programa será julgado segundo a adequação de sua saída à descrição do exercício.
- Seu programa deve consistir de um único arquivo, e deve ser razoavelmente modular, ou seja: deve consistir de diversas funções que, juntas, realizam a tarefa descrita.
- Documente cada função dizendo o quê ela faz.
- Escreva no início do código um cabeçalho com comentários, indicando nome, número do EP, data, nome da disciplina.
- Entregue também um mini-relatório, pode ser em formato txt, explicando como resolveu o exercício, ou seja, como funcionam as diversas partes do seu programa e estimando sua complexidade em função do número de pontos.
- A entrega será eletrônica no ambiente Moodle da disciplina(não receberei exercícios impressos ou via email).