Relatório: Problema de Conexidade 2D

Victor Wichmann Raposo 9298020

Abril 2016

1 Introduction

Esse relatório visa explicar a implementação da minha solição para o problema "Problema de Conexidade 2D". Descreverei, abaixo, os métodos e estruturas utilizadas.
Notação:

- N = número total de pontos ($N \geq 1$)
- D = distância ($D > 0$)

2 Gridding

Com o intuito de aumentar a eficiência do programa, dividi o quadrado unitário em vários quadrados menores de lado D. Dessa maneira, ao ler um novo ponto P basta verificar se ele e D-conexo com os pontos que estejam do mesmo setor da grid e dos setores adjacentes à este. Logo, apenas no pior caso a complexidade do código sera quadrática e não em todos os casos como seria sem essa técnica.

3 ArrayList

Para cada quadrado da grid usei um ArrayList que guarda os pontos contidos nesse setor. Escolhi essa estrutura pois ela é dinâmica e iterável.

4 Point2DIndex

Criei essa classe para representar os pontos recebidos, cada Point2DIndex possui uma coordenada x, uma y e um *index* que é o um inteiro de identificação do ponto.

5 Union Find

A Union Find foi usada para representar o grafo de conexidade entre os N pontos (Point2DIndex), ou seja, se dois pontos P e Q são D-conexos existe um aresta entre eles. Nessa estrutura os pontos são identificados pelos respectivos *index*. Usei a WeightedQuickUnionUF, porque ela tem a melhor complexidade dentre as implementações dessa estrutura.

6 Conclusão

Para solucionar esse problema, fiz uma matriz de $ArrayList < Point2DIndex >$ tal que cada elemento da matriz representa um setor da grid. Cada ponto lido é comparado com os pontos do mesmo setor e dos setores adjacentes, se dois pontos forem D-conexos, uno-os em relação aos índices deles. Ao fim da leitura dos N pontos basta verificar se há apenas uma classe de equivalência, ou seja, se de um ponto qualquer do grafo existe um caminho para todos os outros pontos.