

# Relatório do projeto de Geometria Computacional: Versão discreta em 2D do teorema do sanduíche de presunto

Giovanna Kobus Conrado e Pedro Sousa

NUSP 10262590 e 9793648

## 1 Problema

Dado um conjunto de  $n$  pontos no plano, cada um deles vermelho ou azul, devemos achar uma reta que bissecta, ao mesmo tempo, os pontos vermelhos e os pontos azuis.

Para resolver o problema, porém, consideraremos sua versão no plano dual, ou seja: dado um conjunto de  $n$  retas, cada uma delas vermelha ou azul, achar um ponto tal que, se considerarmos a projeção de todas as retas em sua coordenada  $x$ , temos metade das retas de cada cor acima desse ponto e metade embaixo deste.

## 2 Algoritmo

O algoritmo implementado baseia-se na prova da existência do corte de sanduíche de presunto: sabemos que existe uma intersecção entre os níveis médios dos conjuntos de retas de cada cor, e podemos achá-la por meio de um algoritmo recursivo descrito em [1], que roda em tempo linear.

O algoritmo funciona em fases, e vai recursivamente descartando uma fração de retas até estas serem poucas o suficiente para podermos achar a resposta com um algoritmo de força bruta.

No início de cada fase mantemos:

- Um intervalo aberto  $T$  no eixo  $x$
- Os conjuntos  $G_1$  e  $G_2$  de linhas que estão sendo consideradas
- $p_1$  e  $p_2$ , referentes aos níveis de  $G_1$  e  $G_2$  cuja intersecção quero encontrar.

Irei assumir, sem perda de generalidade, que o conjunto azul é o conjunto com mais retas.

Em cada fase faço uma espécie de "busca binária" em  $T$  de forma a encontrar um intervalo que contenha menos de  $1/32$  do número máximo de intersecções entre retas azuis e ainda contenha a intersecção entre os níveis médios que estamos procurando. Para isso precisamos de duas coisas: contar o número de intersecções dado um intervalo e um conjunto de retas, e uma maneira de encontrar uma intersecção aleatória, ambas em tempo linear.

Depois de achar esse intervalo  $(l, r)$ , construímos um trapezoide usando a intersecção das retas  $x = l$  e  $x = r$  e das retas que estão  $1/8$  acima e abaixo do nível  $p_1$ .

Vamos descartar todas as retas que não intersectam com esse trapezoide e recalculamos os valores de  $p_1$  e  $p_2$ . Depois que essas retas são descartadas e são calculados os novos níveis médios, repetimos esse processo até que tenhamos um número suficientemente pequeno de retas.

Agora, podemos simplesmente achar de maneira *naive* os níveis médios dos conjuntos de retas e sua intersecção, pois como temos um número fixo de retas, a complexidade desse passo será  $O(1)$ . Depois de finalizado o algoritmo, conferimos se a resposta é válida, de forma a capturar eventuais erros no código.

### 3 Restrições

Ao implementar esse algoritmos tivemos alguns problemas.

O principal deles é que não conseguimos contar as intersecções de um conjunto de retas ou achar uma aleatória em tempo linear. O que teríamos que fazer é gerar uma permutação a partir das retas no começo do intervalo considerado e uma no final, e procurar nisso as inversões. Para fazer isso, porém, precisaríamos ordenar as projeções dessas retas, e isso já consome tempo  $O(n \log n)$ : pior que a complexidade linear que queríamos atingir.

Outro problema é que não sabemos tratar alguns casos de contorno, tais como várias retas que se intersectam no mesmo ponto. Além disso, também tivemos que lidar com erros de precisão, principalmente na checagem da resposta.

### 4 Resultados

Rodamos o programa com diversos conjuntos de pontos gerados aleatoriamente e uma resposta sempre foi encontrada.

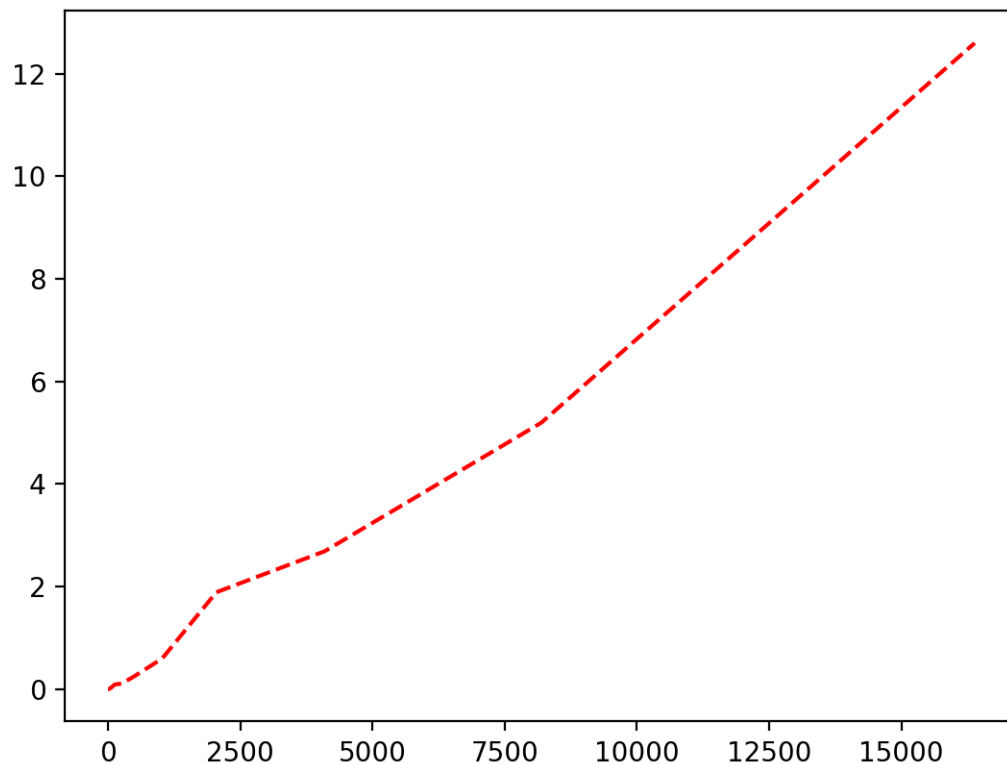


Gráfico de quantidade de pontos (x) e tempo em segundos para achar a resposta (y)

## 5 Referências

1. Chi-Yuan Lo, J. Matoušek and W. Steiger. Discrete Comput. Geom. **11**. 433-452, 1994.
2. Timothy M. Chan and Mihai Pătraşcu. Counting Inversions, Offline Orthogonal Range Counting, and Related Problems. 2010.