

# Localização de Pontos numa Partição do Plano por Segmentos

Joel da Silva Uchoa, Murilo Santos de Lima

DCC – IME – USP

MAC5747 - Geometria Computacional

{joelsu, mslima}@ime.usp.br

## Resumo

Este relatório descreve a implementação de duas estruturas de dados para localização de pontos numa partição do plano por segmentos. Uma das estruturas é construída utilizando uma linha de varredura; a outra utiliza um algoritmo incremental probabilístico. Os resultados experimentais confirmam as análises assintóticas descritas na literatura, de que a segunda estrutura leva menos tempo para ser construída, embora os tempos de consulta sejam semelhantes.

## 1 Introdução

Neste trabalho é descrita a implementação e análise comparativa de duas estruturas de dados para o problema da localização de pontos numa partição do plano por segmentos [1]. A primeira estrutura utiliza um algoritmo de linha de varredura, consome espaço  $O(n^2)$  no pior caso e requer tempo  $O(n^2)$  para ser construída. A segunda é construída utilizando um algoritmo incremental probabilístico, consome espaço esperado  $O(n)$  e tempo esperado  $O(n \lg n)$  para ser construída. Operações de consulta consomem tempo  $O(\lg n)$  em ambas as estruturas, embora na segunda isso seja o tempo esperado. Para uma descrição completa, acompanhada da análise de complexidade das estruturas, ver [1].

Adicionalmente, para fins didáticos, foram desenvolvidas animações para as duas estruturas de dados. Essas animações não são descritas aqui por limitações de espaço.

Na Seção 2 apresenta-se a metodologia de avaliação utilizada, na Seção 3 os resultados experimentais e na Seção 4 as conclusões obtidas a partir das experimentações.

## 2 Metodologia

As estruturas de dados foram implementadas em linguagem Ruby<sup>1</sup>. Ambas foram testadas com três entradas distintas. Foram medidos o número total de passos executados na construção da estrutura e o número total e médio de passos executados numa consulta.

## 3 Resultados experimentais

Nas Tabelas 1 e 2 apresentam-se os resultados experimentais. As colunas  $n$  e  $m$  indicam o número de pontos e segmentos na partição do plano, respectivamente; a coluna  $q$  indica o número de consultas realizadas. Os valores nas análises de tempo consistem no número de operações básicas de comparação realizadas. A Entrada 1 consiste de uma entrada em posição geral, com  $n = 16$ ,  $m = 21$  e  $q = 9$ . A Entrada 2 possui pontos com mesma  $X$ -coordenada e  $n = 19$ ,  $m = 25$  e  $q = 11$ . A Entrada 3 consiste no pior caso para a primeira estrutura de dados, com  $n = 25$ ,  $m = 32$ ; essa entrada não possui consultas, uma vez que seu objetivo é analisar a construção das estruturas.

---

<sup>1</sup><http://www.ruby-lang.org/pt/>

	$n$	$m$	$q$	construção	consulta total	consulta média
Entrada 1	16	21	9	385 passos	80 passos	8,9 passos
Entrada 2	29	25	11	1013 passos	125 passos	11,4 passos
Entrada 3	25	32	0	1761 passos	—	—

Tabela 1: Resultados dos experimentos para a primeira estrutura de dados (linha de varredura).

	$n$	$m$	$q$	construção	consulta total	consulta média
Entrada 1	16	21	9	237 passos	64 passos	7,1 passos
Entrada 2	29	25	11	296 passos	90 passos	8,2 passos
Entrada 3	25	32	0	434 passos	—	—

Tabela 2: Resultados dos experimentos para a segunda estrutura de dados (algoritmo incremental probabilístico).

## 4 Conclusões

Conclui-se que a segunda estrutura de dados possui um desempenho melhor durante a sua construção, e que as duas fornecem tempos de consulta semelhantes, confirmando as análises assintóticas descritas na literatura.

## Referências

- [1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational geometry, algorithms and applications*. Springer Verlag, second edition, 2000.