Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Computational Geometry
Summer term 2018

Exercises 5 - Range Queries[1]

Discussion: Friday, July 6th, 2018

Lecture 11 – Range Queries I (27.06.2018)

Exercise 1 – Worst-Case Running Time. The running-time for range-queries in *kd*-trees is essentially based on the number of regions that must be checked. Let $Q(n)$ be that number.

1. Show that $Q(n)$ is described by the following recurrence:

$$Q(n) = \begin{cases} \mathcal{O}(1) & \text{, für } n = 1 \\ \mathcal{O}(1) + 2Q(n/4) & \text{, für } n > 1 \end{cases}$$

2. Show that $Q(n) = \mathcal{O}(\sqrt{n})$.
3. Show that $\Omega(\sqrt{n})$ is a lower bound for querying in *kd*-trees by defining a set of n points and a query rectangle appropriately.

Hint: For the details of range-queries on *kd*-trees see *Computational Geometry: Algorithms and Applications*

Exercise 2 – Partial Match Queries. *kd*-Trees can be used for *partial match queries*. A 2-dimensional partial match query specifies a value for one of the coordinates and asks for all points that have that value for the specified coordinate. Example: We query all points with x-coordinate 7.

1. Show that 2-dimensional kd-trees can answer partial match queries in $\mathcal{O}(\sqrt{n} + k)$ time, where k is the number of reported points.
2. Explain how to use a 2-dimensional range tree to answer partial match queries. What is the resulting running time?
3. Describe a data structure that uses linear storage and solves partial match queries in $\mathcal{O}(\log n + k)$ time.

Exercise 3 – Range Counting Queries. In some applications one is interested only in the number of points that lie in a range rather than in reporting all of them. Such queries are often referred to as *range counting queries*. In this case one would like to avoid having an additive term of $O(k)$ in the running time.

[1]Based on: M. de Berg, O. Cheong, M. van Kreveld, M. Overmars: Computational Geometry, 3rd ed., Springer-Verlag, 2008.

1. Describe how a 1-dimensional range tree can be adapted such that a range counting query can be performed in $O(\log n)$ time. Prove the query time bound.
2. Describe how to perform d-dimensional range counting queries.

Exercise 4 – Complex Objects. In many applications one wants to apply range queries on complex objects.

1. Let S be a set of n axis-aligned rectangles in the plane. We want to report all rectangles in S that lie in the query rectangle $[x, x'] \times [y, y']$. Describe a data structure that solves this problem using $O(n \log^3 n)$ storage and $O(\log^4 n + k)$ time for a query, where k is the number of reported rectangles.
2. Let P be a set of n polygons in the plane. We want to report all polygons in P that completely lie in the query rectangle $[x, x'] \times [y, y']$. Describe a data structure that solves this problem using $O(n \log^3 n)$ storage and $O(\log^4 n + k)$ time for answering a query, where k is the number of reported polygons.