CMU-CS-78-150

On The Number of Range Queries in k-Space

James B. Saxe
Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

14 November 1978

Abstract

A range query on a set of points in a k-dimensional coordinate space asks for all points lying within a hyperrectangle specified by ranges of permissible values for each of the coordinates. In this paper we regard as identical any two range queries which return the same set of points. We then investigate the number of range queries possible on a set—given a set of N points in k-space, what is the maximum number of distinct subsets that may be specified by giving bounding hyperrectangles. The bounds we find for this number (as a function of N and k) are substantial improvements over previous results, and tighten a lower bound on the time required to process range queries.

This work was supported by the Office of Naval Research under contract N00014-76-C-0370.

Table of Contents

1. Introduction

Given a set (or "file") of points in a k-dimensional coordinate space, a *range query* asks for all points in the set that lie within some hyperrectangle, specified by a range of permissible values for each of the k cooordinates. The *range searching* problem may now be defined as follows: Given a set of N points in k-space, preprocess them so that range queries may be answered quickly. This problem is called "orthogonal range searching" by Knuth (K73, Sec. 6.5).

We concern ourselves here with the number of range queries possible on a set of N points in k-space, where two range queries are considered distinct iff they return different sets of points. We speak of the number of "range queries" rather than the number of "range responses" for two reasons. First, that is the terminology used by Bentley and Maurer [BM78]. Second, our interest in the range searching problem is largely motivated by the more general study of *range-restricted searching problems*. In these problems, a query on a set, S, may be considered to consist of two parts. The first part (or *range restriction*) specifies some hyper-rectangle, R; the second specifies some (arbitrary) query on the set $T = S \cap R$. It is often convenient to partition the possible queries according to the T's selected by their first parts. The number of different T's which may be selected (*i.e.*, the number of equivalence classes of queries) is precisely what we call the number of range queries on S.

It is easy to show that any set of N distinct points on the line admits exactly $\binom{N+1}{2} + 1$ range queries. The answer to a range query is either the empty set or can be defined by two of the N+1 interpoint gaps (including the end spaces). In higher dimensions, the situation is more complicated, since the number of range queries on a set depends not only on the number of points in the set but on their distribution as well. Being interested in worst-case results, we will attempt to determine, given N and k, the *maximum* number of range queries possible on a set of N points in k-space. Bentley and Maurer have shown, for $k \geq 2$, that the maximum number of range queries on N points in k-space lies between the bounds of $(N/2k)^{2k}$

and $N^{2k}/2^k$ (ignoring lower order terms) and they have used this result to show the optimality (within an additive constant) of data structures they call "one level k-ranges". In this paper we shall improve on these bounds. One result of this is to tighten the additive term of Bentley and Maurer's optimality result.

2. A Lower Bound

Consider Figure 1. Here we show N points in the plane divided into two groups, one arranged along the line segment from (1,0) to (0,-1), not including (1,0) or (0,-1), and the other along the line segment from (0,1) to (-1,0), again not including the endpoints. Assume that the two groups are as nearly equal as possible, the first containing $\lceil N/2 \rceil$ points and the second having the remaining $\lfloor N/2 \rfloor$.

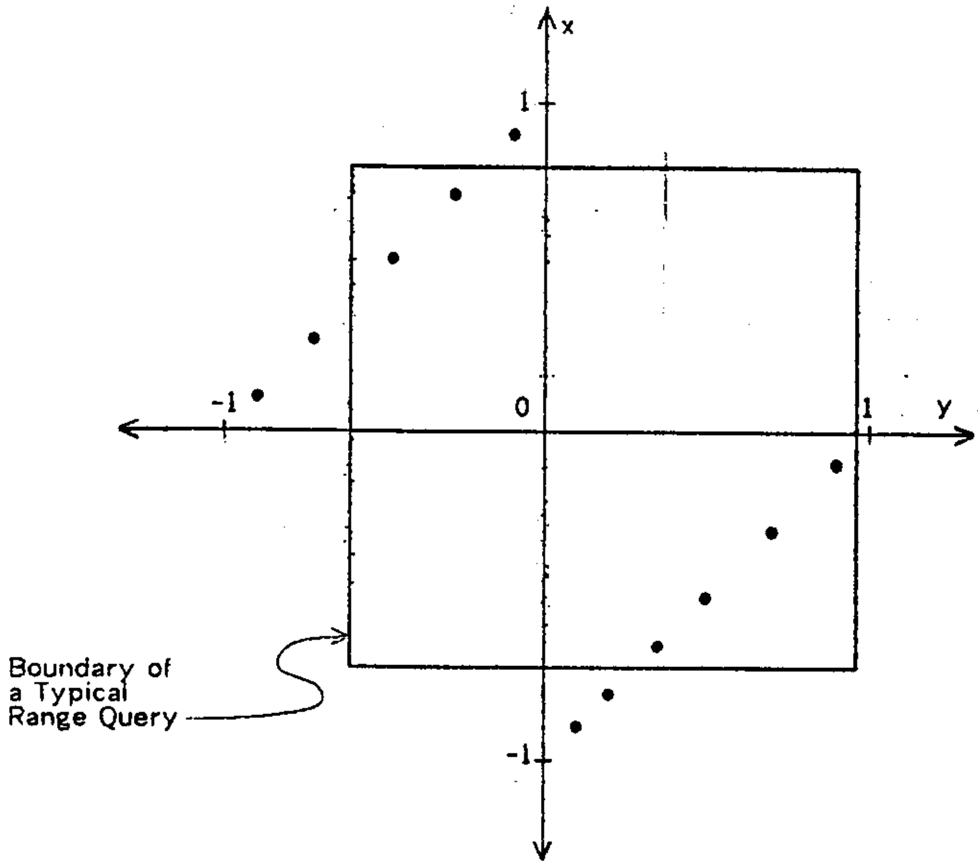

Figure 1: A Distribution Admitting Many Range Queries

We now determine the number of range queries that can be made on this set of points. By adjusting the bottom and right boundaries of the search range, we can select any of

$$\binom{\lceil N/2 \rceil + 1}{2} + 1$$

subsets of the first group to be included in the range (that is, the number of range queries on a set of $\lceil N/2 \rceil$ points in one dimension). Similarly, by adjusting the left and top boundaries, we may include any of

$$\binom{\lfloor N/2 \rfloor + 1}{2} + 1$$

subsets of the second group. Thus, the total number of range queries possible on this set of points is

$$\left[\binom{\lceil N/2 \rceil + 1}{2} + 1\right]\left[\binom{\lfloor N/2 \rfloor + 1}{2} + 1\right]$$
$$\sim [(N/2)^2/2]\,[(N/2)^2/2]$$
$$= N^4/64.$$

Analysis of the lower order terms in the second line of the preceding will show that they are $O(N^3)$ and also that the approximation obtained is conservative to this extent. We have therefore established

Theorem 2-1

The maximum number of range queries on a set of N points in two-space grows at least as $N^4/64$.

The construction of Figure 1 extends naturally to higher-dimensional spaces. In k-space, we divide the N points into k approximately equal groups and arrange them along the k line segments:

from (1,0,0,...,0,0) to (0,-1,0,...,0,0);
from (0,1,0,...,0,0) to (0,0,-1,...,0,0);
. . .
from (0,0,0,...,1,0) to (0,0,0,...,0,-1);
from (0,0,0,...,0,1) to (-1,0,0,...,0,0).

These configurations of points offer a constructive proof of

Theorem 2-2

Let k be a positive integer. Then the maximum number of range queries on a set of N points in k-space grows at least as $N^{2k}/(2^k k^{2k})$.

3. An Upper Bound

The question now arises as to how close the constructions of Section 2 come to achieving the maximum number of range queries. A partial answer to this question is the following result.

Theorem 3-1:

The maximum number of range queries possible on a set of N points in two-space grows no faster than $N^4/48 + O(N^3)$.

Proof:

Consider a set, Y, of N points in the plane. For the purpose of investigating the number of possible range queries on Y, we assume, without loss of generality, that Y is a 1-1 function from {1,2,...,N} onto {1,2,...,N}.[1] Such a set is exhibited in Figure 2.

For each non-empty range query, Q, there is a unique <u>minimal enclosing rectangle</u>, namely $[a,b]\times[c,d]$ where

$$\begin{aligned} a &= \min(\text{domain}(Q)); \\ b &= \max(\text{domain}(Q)); \\ c &= \min(\text{range}(Q)); \\ d &= \max(\text{range}(Q)). \end{aligned}$$

Consider two integers, a and b, with $1 \le a \le b \le N$. How many pairs (c,d) may exist such that $[a,b]\times[c,d]$ is the minimal enclosing rectangle of some range query? Consider the example in Figure 2. We take $c' = \min\{Y(a),Y(b)\}$ and

[1] Here we use the formal definition of a point in two-space as an ordered pair and of a function as a set of ordered pairs.

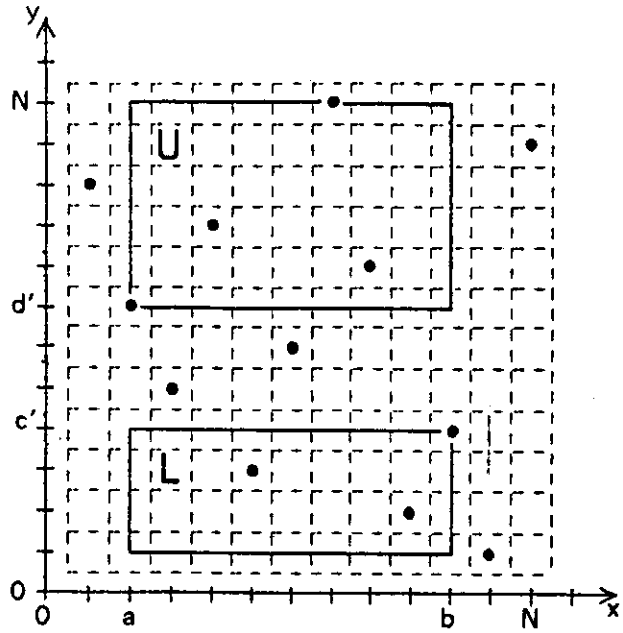

Figure 2: A 1-1 Function from {1,2,...,N} to {1,2,...N}, Viewed as a Point Set in Two-Space

$d' = \max\{Y(a),Y(b)\}$. Then, $[a,b]\times[c,d]$ is a minimal enclosing rectangle iff c is the ordinate of some point of Y which lies in the rectangle $L = [a,b]\times[1,c']$ and d is the ordinate of some point of Y which lies in the rectangle $U = [a,b]\times[d',N]$. Thus, the total number of (c,d) giving rise to minimal enclosing rectangles is $|L\cap Y|\cdot|U\cap Y|$. Since $|L\cap Y|+|U\cap Y|$ can be at most $b-a+1+\delta_{a,b}$,[1] it follows that $|L\cap Y|\cdot|U\cap Y|$ is at most $(\lceil(b-a+1+\delta_{a,b})/2\rceil)(\lfloor(b-a+1+\delta_{a,b})/2\rfloor)$.[2] By summing over all possible values of a and b, we see that the total number of range queries, including the empty query, is no more than

$$1 + \sum_{1\le a\le b\le N} (\lceil(b-a+1+\delta_{a,b})/2\rceil)(\lfloor(b-a+1+\delta_{a,b})/2\rfloor)$$

$$\sim \sum_{1\le a\le b\le N} ((b-a)/2)^2$$

---

[1] δ here signifies the Kroenecker δ-function.

[2] Since given the sum of two integers, their product is maximized by making them as nearly equal as possible.

$$= \sum_{1 \le b \le N} \left(\sum_{1 \le a \le b} ((b-a)/2)^2 \right)$$

$$\sim \sum_{1 \le b \le N} b^3/12$$

$$\sim N^4/48.$$

If the approximations made in the preceding calculation are studied, it will be seen that the error introduced is at most $O(N^3)$. This completes the proof. □

We can extend the previous result by induction to give bounds on the number of range queries in higher dimensions as follows:

Theorem 3-2:

Let k be an integer greater than unity. Then the maximum number of range queries on a set of N points in k-space grows no faster than $N^{2k}/(2\cdot(2k)!) + O(N^{2k-1})$.

Proof:

By Theorem 3-1, the result holds for the case where k=2. Thus we need only prove the result for k>2, assuming the result for k-1.

Consider a set, Y, of N points in k-space. Without loss of generality, we assume the kth coordinates of the points in Y to be precisely the integers 1,...,N. Thus, each range query on Y may be expressed by giving two integers, a and b, (with $1 \le a \le b \le N$) bounding the query set in the kth coordinate, together with the specification of a (k-1)-dimensional range query on b-a+1 points whose kth coordinates lie in the closed interval [a,b]. Using the notation $R_j(M)$ to represent the maximum number of range queries possible on any set of M points in j-space, we now have that

$$R_k(N) \le \sum_{1 \le a \le b \le N} R_{k-1}(a-b+1).$$

$$\sim \sum_{1 \le b \le N} \sum_{1 \le c \le b} R_{k-1}(c)$$

$$\sim \sum_{1 \le b \le N} \sum_{1 \le c \le b} c^{2k-2}/(2\cdot(2k-2)!)$$

$$\sim \sum_{1 \le b \le N} b^{2k-1}/(2\cdot(2k-1)!)$$

$$\sim N^{2k}/(2\cdot(2k)!).$$

As before, careful analysis of the lower-order terms will show them to be $O(N^{2k-1})$. □

4. Conclusions

The bounds given in Sections 2 and 3 for the maximum number of range queries in two dimensions tighten the results of Bentley and Maurer considerably--from a factor of 64 difference between the lower and upper bounds to a factor of 4/3. Similar improvements are obtained in higher dimensions, though the final results there are still looser than for the two-dimensional case. The results for the two- and three-dimensional cases are summarized in the following table.

Dimension of Space	Bentley & Maurer			New Results		
	Lower Bound	Upper Bound	Ratio	Lower Bound	Upper Bound	Ratio
2	$N^4/256$	$N^4/4$	64	$N^4/64$	$N^4/48$	4/3
3	$N^6/46656$	$N^6/8$	5832	$N^6/5832$	$N^6/1440$	4.05

It is clear that any decision-tree program for range searching must use at least as many comparisons as the logarithm to the base two of the number of possible responses. Bentley and Maurer coupled this fact with their lower bound on the number of range queries to show a lower bound on the worst case complexity of range searching of $\log_2(N^4/256) = 4 \log_2 N - 8$. Our lower bound tightens their result to $4 \log_2 N - 6$, and our upper bound shows that this method cannot be used to decrease the additive constant much further.

Similar results are obtained for higher-dimensional spaces. For k-dimensional space, we get a lower bound of $2k \log_2 N - k(1+2 \log_2 k)$ and our upper bound shows that the decision-tree argument cannot be used to give a lower bound greater than $2k \log_2 N - k(1+\log_2 k-\log_2 e) - 3(\log_2(\pi k))/2$. This last result is obtained by using Stirling's approximation to estimate the value of $(2k)!$, which appears in Theorem 3-2.

The most obvious open problem left by this work is that of further tightening the bounds. The author suspects (but will not bet money) that the lower bounds given in Section 2 may be exact up to second-order terms; at any rate, the upper bounds of Section 3 are computed on the basis of very optimistic assumptions. The bound of Theorem 3-1, for example, could be exactly achieved only if N points could be placed in the plane so that no one of them lay within the minimal enclosing rectangle of any other two. This last condition, however, is impossible to achieve for $N>4$.

A deeper problem is that of studying the *structure*, rather than just the cardinality, of sets of all possible range queries over a (given) set of points in k-space. In particular, the complexity of range searching in k-space appears to depend on the dimension of the space to an extent not entirely accounted for by the sheer number of possible range searches. To give an example, there are $O(N^2)$ range queries on a set of N points in one dimension. By storing the points as a sorted list, it becomes possible to answer range queries in $O(\lg N)$ time (plus reporting time proportional to the number of points actually in the range). The preprocessing time required is $O(N \lg N)$ total, or $O(\lg N)$ per point. Consider on the other hand a set of $N^{1/2}$ points in two-space. The number of possible queries is again $O(N^2)$ (in fact, the constant term is smaller for this case). But now, if we allow only $O(\lg N)$ preprocessing per point, the best known algorithm [BF78] requires $O(\lg^2 N)$ time (plus reporting time) to answer a range query. To take another example, range searching on one of the distributions constructed in Theorem 2-2 is very simple (if such a distribution is expected in advance), since a k-dimensional range query on such a set can be reduced to k one-dimensional queries. Random distributions of points in k-space, while they may admit many fewer distinct range queries than the sets of Theorem 2-2, require either more preprocessing or more query time, at least using currently known algorithms. By seeking a deeper understanding of these phenomena, we may hope to shed light not only on the range searching problem, but on more general range-restricted searching problems as well.

References

[BF78] Bentley, J. L. and J. H. Friedman. A Survey of Algorithms and Data Structures for Range Searching, in Two Papers on Range Searching, J. L. Bentley, J. H. Friedman, and H. A. Maurer. Carnegie-Mellon University Computer Science Department Report CMU-CS-78-136. August, 1978.

[BM78] Bentley, J. L. and H. A. Maurer. Efficient Worst-Case Data Structures for Range Searching, in Two Papers on Range Searching, J. L. Bentley, J. H. Friedman, and H. A. Maurer. Carnegie-Mellon University Computer Science Department Report CMU-CS-78-136. August, 1978.

[K73] Knuth, D. E. *Sorting and Searching. The Art of Computer Programming.* Vol. 3. Addison-Wesley. Reading, Massachusetts. 1973.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER CMU-CS-78-150	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ON THE NUMBER OF RANGE QUERIES IN k-SPACE		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James B. Saxe		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0370
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Department Schenley Park, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE November 14, 1978
		13. NUMBER OF PAGES 13
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same as above		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)