Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The range 1 query (R1Q) problem

Michael A. Bender [a,b], Rezaul A. Chowdhury [a], Pramod Ganapathi [a,*], Samuel McCauley [a], Yuan Tang [c]

[a] Department of Computer Science, Stony Brook University, NY, USA
[b] Tokutek, Inc., USA
[c] Software School, and, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, China

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form 27 August 2015
Accepted 30 December 2015
Available online xxxx

Keywords:
Range 1 query
R1Q
Range query
Range emptiness
Emptiness query
Existential query
Bit matrix
Randomized
Rectangular
Orthogonal
Non-orthogonal
Triangular
Polygonal
Simplicial
Circular
Spherical

ABSTRACT

We define the ***range 1 query (R1Q) problem*** as follows. Given a d-dimensional ($d \geq 1$) input bit matrix A (consisting of 0's and 1's), preprocess A so that for any given region $\mathcal{R}$ of A, efficiently answer queries asking if $\mathcal{R}$ contains a 1 or not. We consider both orthogonal and non-orthogonal shapes for $\mathcal{R}$ including rectangles, axis-parallel right-triangles, certain types of polygons, axis-parallel right simplices and spheres. We provide space-efficient deterministic and randomized algorithms with constant query times (in constant dimensions) for solving the problem in the word-RAM model. The space usage in bits is sublinear, linear, or near linear in the size of A, depending on the algorithm.

1. Introduction

Range searching is one of the most fundamental problems in computational geometry. Many geometric problems can be modeled as range searching problems. It arises in application areas including geographical information systems, computer graphics, computer aided design, spatial databases, and time series databases. Range searching encompasses different types of problems, such as range counting, range reporting, emptiness queries, and optimization queries. This problem has been extensively studied in literature [1–5].

The ***range 1 query (R1Q) problem*** is defined as follows. Given a d-dimensional ($d \geq 1$) input bit matrix A (consisting of 0's and 1's), preprocess A to efficiently answer queries asking if any given range $\mathcal{R}$ of A is empty (does not contain a 1)

* Corresponding author.
E-mail addresses: bender@cs.stonybrook.edu (M.A. Bender), rezaul@cs.stonybrook.edu (R.A. Chowdhury), pganapathi@cs.stonybrook.edu (P. Ganapathi), smccauley@cs.stonybrook.edu (S. McCauley), yuantang@fudan.edu.cn (Y. Tang).

http://dx.doi.org/10.1016/j.tcs.2015.12.040
0304-3975/

or not, denoted by $\mathrm{R1Q}_A(\mathcal{R})$ or simply $\mathrm{R1Q}(\mathcal{R})$. For example, in 2-D, the range $\mathcal{R}$ can be a rectangle, a right triangle, a polygon, a circle, or any other shape.

In this paper, we investigate solutions to the R1Q problem in the ***word-RAM model***. If N is the input size in bits, the machine word is assumed to be of size $\Omega(\log N)$ bits. We assume that accesses to memory cells, arithmetic operations $(+, -, \times, \div, \mathrm{mod})$, comparisons $(<, >, =, \leq, \geq)$, bitwise operations (bitwise-AND, OR, XOR), and shifting a word left or right a specific number of bits each take $\Theta(1)$ time. Almost all our solutions can be easily implemented to use only bit shifts, bitwise ORs, comparisons, and memory accesses in the query execution algorithms. Our solutions are amenable to hardware that do not use complicated arithmetic operations i.e., additions, subtractions, multiplications, divisions, and mod.

The solutions we investigate share the following characteristics, if possible. First of all, we want queries to run in constant time (assuming d is fixed), even for $d \geq 2$ dimensions. Second, we are interested in solutions that have space linear or sublinear in the number of bits in the input grid. Depending on the algorithm, we give solutions in one of the two models [6]: (*a*) indexing model (or systematic scheme), in which we retain the input matrix to be accessed by the query execution algorithms, and space generally refers to the additional space for the index data structure; or (*b*) encoding model (or non-systematic scheme), in which we discard the input matrix and access only the preprocessed data; Here, space refers to the total space. Note that while our sublinear bounds are parameterized by the number of 1's in the grid, this is still larger than the information-theoretic lower bounds. For our motivating applications, information-theoretically optimal space is less important than constant query time. Third, we are interested in grid inputs [7,8], where the input is a matrix consisting of 0's and 1's. Hence we view the problem in terms of pixels/voxels rather than as a set of spatial points. We take advantage of this grid perspective in our algorithms using table lookups and hashing to achieve constant-time queries. Finally, we are interested in both orthogonal and non-orthogonal queries, and we require solutions that are simple enough to be implementable.

1.1. Previous results

The R1Q problem can be solved using data structures such as balanced binary search trees, kd-trees, quad trees, range trees, partition trees, and cutting trees (see [1]), which take the positions of the 1-bits as input. It can also be solved using a data structure proposed by Overmars [7], which uses priority search trees, y-fast tries, and q-fast tries and takes the entire grid as input. However, in d-D ($d \geq 2$), in the worst case, these data structures have a query time at least polylogarithmic in the number of 1's in the grid and occupy a near-linear number of bits w.r.t. the number of 1's.

The R1Q problem can also be solved via the solutions of the range partial sum [9,10] and the range minimum query (RMQ) [11–21] problems. Though several efficient algorithms have been developed to solve the problem in 1-D and 2-D, their generalizations to 3-D and higher dimensions (while occupying a linear number of bits) are not known yet. Also, there is little work on space-efficient constant-time RMQ solutions for non-orthogonal ranges. Also, some of the above data structures that use trees and partial sum use semigroup arithmetic model, which is more restrictive than our model and hence the results cannot be compared fairly.

The R1Q problem can also be solved using rank queries. The data structures to answer rank queries in [22–31] either do not generalize to 2-D and higher dimensions or they take super-constant time for answering queries. Farzan et al. [32, 33] present space-efficient data structures to answer orthogonal rank queries in constant time for all dimensions. They present simple linear-space structures as well as more complicated structures occupying entropy-bounded space. Their data structures are more space-efficient than our orthogonal deterministic structures and answer the more general problem of rank queries.

There are two major differences between our solution and Farzan et al.'s method to solve the R1Q problem. First and foremost, the powers-of-2 approach that we use to answer orthogonal R1Q deterministically can be easily extended to answer orthogonal R1Q probabilistically using randomized algorithms in the encoding model. If we allow some errors, we can reduce the space to significantly sublinear size when the number of 1-bits is asymptotically smaller than the total number of bits, while answering the queries in constant time. The powers-of-2 approach can also be used to answer right-triangular R1Q (which forms the basis for some polygonal queries) in constant time, occupying near-linear space in bits, which is significantly less for right-triangular queries compared to the space consumption of standard approaches. On the other hand, it is not clear how Farzan et al.'s method can be extended to answer orthogonal R1Q probabilistically using sublinear space or to answer right-triangular R1Q. Second, almost all our query execution algorithms can be made to use the following four simple operations only: comparisons, bit shifts, bitwise ORs, and memory accesses (see Section 2) without asymptotic increase in space and time bounds. Hence they can be implemented on a simpler hardware without complicated arithmetic operations. However, Farzan et al.'s method uses additions, subtractions, and more expensive operations such as multiplications on the values. It is not straightforward to replace all those arithmetic operations that work on values with simpler non-arithmetic operations while retaining the linear-space and constant-time bounds.

1.2. Motivation

We encountered the R1Q and R0Q (whether a range contains a 0) problems while trying to optimize stencil computations [34] in the ***Pochoir*** stencil compiler [35,36], where we had to answer octagonal R1Q and octagonal R0Q on a static 2-D

Please cite this article in press as: M.A. Bender et al., The range 1 query (R1Q) problem, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2015.12.040