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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Simultaneous encodings for range and next/previous larger/smaller value queries ☆

Seungbum Jo, Srinivasa Rao Satti *

Seoul National University, South Korea

A R T I C L E I N F O

Article history:
Received 2 November 2015
Received in revised form 8 January 2016
Accepted 28 January 2016
Available online xxxx

Keywords:
Range minimum queries
Next/previous larger values
2*d*-Min heap
Encoding
Balanced parenthesis sequence

A B S T R A C T

Given an array of n elements from a total order, we propose encodings that support various range queries (range minimum, range maximum and their variants), and previous and next smaller/larger value queries. When query time is not of concern, we obtain a $4.088n + o(n)$-bit encoding that supports all these queries. For the case when we need to support all these queries in constant time, we give an encoding that takes $4.585n + o(n)$ bits, where n is the length of input array. This improves the $5.08n + o(n)$-bit encoding obtained by encoding the colored $2d$-Min and Max heaps proposed by Fischer [11]. We first extend the original DFUDS [8] encoding of the colored 2d-Min (Max) heap that supports the queries in constant time. Then, we combine the extended DFUDS of $2d$-Min heap and $2d$-Max heap using the Min–Max encoding of Gawrychowski and Nicholson [15] with some modifications. We also obtain encodings that take lesser space and support a subset of these queries.

1. Introduction

Given an array $A[1 \ldots n]$ of n elements from a total order. For $1 \leq i \leq j \leq n$, suppose that there are m (l) positions $i \leq p_1 \leq \ldots \leq p_m \leq j$ ($i \leq q_1 \leq \ldots \leq q_l \leq j$) in A which are the positions of minimum (maximum) values between $A[i]$ and $A[j]$. Then we can define various range minimum (maximum) queries as follows.

- Range Minimum Query ($\mathsf{RMinQ}_A(i, j)$): Return an arbitrary position among $p_1, \ldots, p_m$.
- Range Leftmost Minimum Query ($\mathsf{RLMinQ}_A(i, j)$): Return p_1.
- Range Rightmost Minimum Query ($\mathsf{RRMinQ}_A(i, j)$): Return p_j.
- Range k-th Minimum Query ($\mathsf{RkMinQ}_A(i, j)$): Return p_k (for $1 \leq k \leq m$).
- Range Maximum Query ($\mathsf{RMaxQ}_A(i, j)$): Return an arbitrary position among $q_1, \ldots, q_l$.
- Range Leftmost Maximum Query ($\mathsf{RLMaxQ}_A(i, j)$): Return q_1.
- Range Rightmost Maximum Query ($\mathsf{RRMaxQ}_A(i, j)$): Return q_l.
- Range k-th Maximum Query ($\mathsf{RkMaxQ}_A(i, j)$): Return q_k (for $1 \leq k \leq l$).

Also for $1 \leq i \leq n$, we consider following additional queries on A.

☆ Preliminary version of these results have appeared in the proceedings of the 21st International Computing and Combinatorics Conference (COCOON-2015) [1].

* Corresponding author.
E-mail addresses: sbcho@tcs.snu.ac.kr (S. Jo), ssrao@cse.snu.ac.kr (S.R. Satti).

http://dx.doi.org/10.1016/j.tcs.2016.01.043
0304-3975/

Please cite this article in press as: S. Jo, S.R. Satti, Simultaneous encodings for range and next/previous larger/smaller value queries, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.01.043

- Previous Smaller Value ($\mathsf{PSV}_A(i)$): $\max(j : j < i, A[j] < A[i])$.
- Next Smaller Value ($\mathsf{NSV}_A(i)$): $\min(j : j > i, A[j] < A[i])$.
- Previous Larger Value ($\mathsf{PLV}_A(i)$): $\max(j : j < i, A[j] > A[i])$.
- Next Larger Value ($\mathsf{NLV}_A(i)$): $\min(j : j > i, A[j] > A[i])$.

For defined above four queries formally, we assume that $A[0] = A[n+1] = -\infty$ for $\mathsf{PSV}_A(i)$ and $\mathsf{NSV}_A(i)$. Similarly we assume that $A[0] = A[n+1] = \infty$ for $\mathsf{PLV}_A(i)$ and $\mathsf{NLV}_A(i)$.

Our aim is to obtain space-efficient encodings that support these queries efficiently. An encoding should support the queries without accessing the input array (at query time). The minimum size of an encoding is also referred to as the *effective entropy* of the input data (with respect to the queries) [2]. We assume the standard word-RAM model [3] with word size $\Theta(\lg n)$.

Previous work The range minimum/maximum problem has been well-studied in the literature. It is well-known [4] that finding RMinQ_A can be transformed to the problem of finding the LCA (Lowest Common Ancestor) between (the nodes corresponding to) the two query positions in the Cartesian tree constructed on A. Furthermore, since different topological structures of the Cartesian tree on A give rise to different set of answers for RMinQ_A on A, one can obtain an information-theoretic lower bound of $2n - \Theta(\lg n)$[1] bits on the encoding of A that answers RMinQ queries. Sadakane [5] proposed the $4n + o(n)$-bit encoding with constant query time for RMinQ_A problem using the balanced parentheses (BP) [6] of the Cartesian tree of A with some additional nodes. Fischer and Heun [7] introduced the 2d-Min heap, which is a variant of the Cartesian tree, and showed how to encode it using the Depth first unary degree sequence (DFUDS) [8] representation in $2n + o(n)$ bits which supports RMinQ_A queries in constant time. Davoodi et al. show that same $2n + o(n)$-bit encoding with constant query time can be obtained by encoding the Cartesian trees [9]. For RkMinQ_A, Fischer and Heun [10] defined the *approximate range median of minima query* problem which returns a position RkMinQ_A for some $\frac{1}{16}m \le k \le \frac{15}{16}m$, and proposed an encoding that uses $2.54n + o(n)$ bits and supports the *approximate* RMinQ_A queries in constant time, using a *Super Cartesian tree*.

For PSV_A and NSV_A, if all elements in A are distinct, then $2n + o(n)$ bits are enough to answer the queries in constant time, by using the 2d-Min heap of Fischer and Heun [7]. For the general case, Fischer [11] proposed the *colored* 2d*-Min heap*, and proposed an optimal $2.54n + o(n)$-bit encoding which can answer PSV_A and NSV_A in constant time. As the extension of the PSV_A and NSV_A, one can define the *Nearest Larger Neighbor(*$\mathsf{NLN}(i)$*)* on A which returns $\mathsf{PSV}_A(i)$ if $i - \mathsf{PSV}_A(i) \le \mathsf{NSV}_A(i) - i$ and returns $\mathsf{NSV}_A(i)$ otherwise. This problem was first discussed by Berkman et al. [12] and they proposed a parallel algorithm to answer NLN queries for all positions on the array (this problem is defined as *All-Nearest Larger Neighbor (*ANLN*) problem.*) and Asano and Kirkpatrick [13] proposed time–space tradeoff algorithms for ANLN problem. Jayapaul et al. [14] proposed $2n + o(n)$-bit encoding which supports an $\mathsf{NLN}(i)$ on A in constant time if all elements in A are distinct.

One can support both RMinQ_A and RMaxQ_A in constant time trivially using the encodings for RMinQ_A and RMaxQ_A queries, using a total of $4n + o(n)$ bits. Gawrychowski and Nicholson reduce this space to $3n + o(n)$ bits while maintaining constant time query time [15]. Their scheme also can support PSV_A and PLV_A in constant time when there are no consecutive equal elements in A.

Our results In this paper, we first extend the original DFUDS [8] for colored 2d-Min(Max) heap that supports the queries in constant time. Then, we combine the extended DFUDS of 2d-Min heap and 2d-Max heap using Gawrychowski and Nicholson's Min–Max encoding [15] with some modifications. As a result, we obtain the following non-trivial encodings that support a wide range of queries.

Theorem 1. *An array $A[1 \dots n]$ containing n elements from a total order can be encoded using*

(a) *at most* $3.17n + o(n)$ *bits to support* RMinQ_A, RMaxQ_A, RRMinQ_A, RRMaxQ_A, PSV_A, *and* PLV_A *queries;*
(b) *at most* $3.322n + o(n)$ *bits to support the queries in (a) in constant time;*
(c) *at most* $4.088n + o(n)$ *bits to support* RMinQ_A, RRMinQ_A, RLMinQ_A, RkMinQ_A, PSV_A, NSV_A, RMaxQ_A, RRMaxQ_A, RLMaxQ_A, RkMaxQ_A, PLV_A *and* NLV_A *queries; and*
(d) *at most* $4.585n + o(n)$ *bits to support the queries in (c) in constant time.*

If the array contains no two consecutive equal elements, then (a) and (b) take $3n + o(n)$ *bits, and (c) and (d) take* $4n + o(n)$ *bits.*

This paper organized as follows. Section 2 introduces various data structures that we use later in our encodings. In Section 3, we describe the encoding of colored 2d-Min heap by extending the DFUDS of 2d-Min heap. This encoding uses a distinct approach from the encoding of the colored 2d-Min heap by Fischer [11]. Finally, in Section 4, we combine the encoding of this colored 2d-Min heap and Gawrychowski and Nicholson's Min–Max encoding [15] with some modifications, to obtain our main result (Theorem 1).

[1] We use $\lg n$ to denote $\log_2 n$.