Consider the following problem.

**Range Minimum Query, RMQ** Given an array $A[1..n]$ and a range $[s, t] \subset [1, n]$, a range minimum query asks the position of the minimum value in $A[s..t]$. If there exist more than one minimum values in the query range, return the leftmost one.

We consider the indexing problem, that is, given the array $A$, we first construct a data structure $D_A$, then given a query range, we solve the problem using $D_A$. There exists a linear space (O($n$) words) data structure for the RMQ problem supporting constant time queries [3,4]. It is however complicated and there have been no efficient implementations until recently. In 2000, a simple solution [1] was given and after that, constant query time RMQ data structures are used in many algorithms.

In this talk, we explain an $O(n)$-word data structure for the RMQ problem. Then we reduce the size of the data structure to just $2n + o(n)$ bits [2]. We also explain applications of the problem such as compressed suffix trees [5].

**References**


