# On multidimensional range queries

Tuukka Korhonen

October 30, 2019

The following question is frequently asked in Codeforces: *Is there a 2D segment tree that supports range addition and range minimum?* In this document I give evidence that such data structure does not exists, or if it would exists it would not generalize to higher dimensions. In particular I show that if for all $d$ a $d$-dimensional data structure that performs such queries in $O(polylog(N))$ time would exist, then the exponential time hypothesis would fail. Such data structure exists for range addition and range sum, so this is a non-trivial claim separating the hardness of these problems.

## 1 Formalization

Consider a $d$-dimensional integer array $A$ whose elements are indexed with $A[(i_1, \dots, i_d)]$ for all $0 \leq i_j < N$. Furthermore assume that $A$ is initialized to zero. The operation $add(v, l_1, r_1, \dots, l_d, r_d)$ adds the value $v$ to all elements $A[(i_1, \dots, i_d)]$ such that $l_j \leq i_j < r_j$. The operation $min(l_1, r_1, \dots, l_d, r_d)$ returns the minimum value over all elements $A[(i_1, \dots, i_d)]$ such that $l_j \leq i_j < r_j$. A data structure that supports these two operations is called Range-Add-Min-structure.

In the Boolean satisfiability problem (SAT) the task is to decide if there is an assigment of $n$ binary variables to true/false so that it satisfies a given Boolean formula. The formula consists of $m$ *clauses*, and an assigment of variables satisfies the formula if it satisfies every clause. A clause is a logical or that consists of variables and negations of variables. For example if $x_1, x_2, \dots, x_n$ are variables, the clause $(x_1 \vee \neg x_3 \vee x_6)$ is satisfied if $x_1$ is true or $x_3$ is false or $x_6$ is true.

The problem $k$-SAT is Boolean satisfiability problem where each clause has at most $k$ variables or negations of variables. 2-SAT can be solved in polynomial time, but 3-SAT is NP-hard. Furthermore a widely believed *exponential time hypothesis* conjectures that there exists $\varepsilon > 0$ such that 3-SAT cannot be solved in time $O((1+\varepsilon)^n poly(n, m))$, where $poly(n, m)$ is polynomial function on $n$ and $m$.

**Theorem 1.** *If there exists a function $f(d)$ such that $d$-dimensional* Range-Add-Min-*structure can process $Q$ operations in $O((\log_2(N) + \log_2(Q))^{f(d)} Q)$ time for all $d$, then exponential time hypothesis fails.*

## 2 Proof

We prove Theorem 1 by assuming that such $d$-dimensional data structure exists, and constructing a $O(2^{3n/d} poly(n, m))$ time algorithm for 3-SAT, where $n$ is the number of variables and $m$ is the number of clauses.

**Claim 1.** *For all $k$ and $d$, $k$-SAT can be solved with $O(2^{kn/d} m)$ operations with a $d$-dimensional* Range-Add-Min-*structure with $N = 2^{n/d}$.*

*Proof.* Suppose without loss of generality that $d$ divides $n$. Let $N = 2^{n/d}$. We use the $d$-dimensional array $A$ with indices $(i_1, \ldots, i_d)$, $0 \leq i_j < N$. A has $2^n$ elements, because $N^d = (2^{n/d})^d = 2^n$.

Let $x_0, \ldots, x_{n-1}$ be a bit string of length $n$. The bijection $\phi$ with

$$\phi(x_0, \ldots, x_{n-1}) = (1x_0 + 2x_1 + 4x_2 + \ldots + x_{n/d-1}2^{n/d-1}, \ldots, 1x_{n-n/d} + \ldots)$$

maps bit strings of length $n$ to the indices of $A$. For example if $n = 6$ and $d = 2$, the bijection is $\phi(x_0, x_1, x_2, x_3, x_4, x_5) = (1x_0 + 2x_1 + 4x_2, 1x_3 + 2x_4 + 4x_5)$.

An assignment of variables in SAT is a bit string of length $n$. We use the array $A$ to represent if the assigment $x_0, x_1, \ldots, x_{n-1}$ is a satisfiable assignment of the SAT formula, using the bijection $\phi(x_0, \ldots, x_{n-1})$ to map between variable assignments and indices of $A$. We iteratively add clauses to the formula, maintaining the invariant that $A[\phi(x_0, \ldots, x_{n-1})] = 0$ if and only if the assigment $x_0, \ldots, x_{n-1}$ satisfies all clauses added so far.

**Claim 2.** *Given a clause that contains $k$ variables or negations of variables, we can add 1 to all elements $A[\phi(x_0, \ldots, x_{n-1})]$ such that $x_0, \ldots, x_{n-1}$ does not satisfy the clause using at most $2^{kn/d}$ add-operations.*

*Proof.* Let $v_1, \ldots, v_k$ be the variables that occur in the clause (possibly with negations). Only the values $x_{v_1}, \ldots, x_{v_k}$ affect whether the clause is satisfied or not. These values affect at most $k$ dimensions in the mapping $\phi(x_0, \ldots, x_{n-1})$. Lets call these dimensions the non-trivial dimensions, and other dimensions the trivial dimensions. If we choose the values of non-trivial dimensions, then by the inverse mapping $\phi^{-1}$ we choose the values of $x_{v_1}, \ldots, x_{v_k}$, and know if the clause is satisfied or not. We brute force over all possible values of non-trivial dimensions. For each value, if the clause is not be satisfied by $\phi^{-1}$, we use operation $add(1, l_1, r_1, \ldots, l_d, r_d)$, where $l_j = r_j - 1$ is the brute forced value of dimension $j$ if $j$ is a non-trivial dimension and otherwise $l_j = 0$ and $r_j = N$. Each dimension has $N$ values, so brute forcing the non-trivial dimensions takes at most $N^k = (2^{n/d})^k = 2^{kn/d}$ *add*-operations. □

By Claim 2 each clause can be processed with $2^{kn/d}$ *add*-operations. Therefore when adding all $m$ clauses we use at most $m2^{kn/d}$ *add*-operations. After all clauses have been added, we use a single *min*-operation where $l_j = 0$ and $r_j = N$ for all dimensions $j$ to query if there are any elements in $A$ that have value 0, and thus correspond to satisfiable assignments. □

**Claim 3.** *If the data structure of Theorem 1 exists, k-SAT can be solved in $O((1+\varepsilon)^n poly(n,m))$ for any $\varepsilon > 0$.*

*Proof.* Given $k$ and $\varepsilon$, lets choose $d$ so that $2^{k/d} < 1+\varepsilon$. By using the algorithm of Claim 1 with the data structure of Theorem 1 we obtain a runtime of

$$\begin{aligned} &O((\log_2(2^{n/d}) + \log_2(2^{kn/d}m))^{f(d)} 2^{kn/d} m) \\ =&O((n/d + kn\log_2(m)/d)^{f(d)} 2^{kn/d} m) \\ =&O((1+\varepsilon)^n poly(n,m)). \end{aligned}$$

□

Claim 3 completes the proof, because the exponential time hypothesis states that there exists $\varepsilon > 0$ such that no $O((1+\varepsilon)^n poly(n,m))$ algorithm exists for 3-SAT.

## 3 Conclusion

We proved that efficient $d$-dimensional Range-Add-Min-structure does not exist if the exponential time hypothesis holds. Our proof allows the existance of such 2D data structure. However we know that such 2D data structure must be developed with techniques that do not generalize to higher dimensions. An 8-dimensional Range-Add-Min-structure would imply a state-of-the-art algorithm for 3-SAT trough our reduction. Some open questions are:

1. Is there a conditional hardness proof for 2D case?
2. Likewise to $(min,+)$-semiring, our reduction also works for $(+,\cdot)$-semiring. Does it work for all commutative semirings?
3. A $d$-dimensional data structure with $O(polylog(N))$ operations exists if the operations are addition and sum. Can we classify all pairs of operations so that the data structure either exists, or does not exist assuming ETH?
4. Is it plausible that techniques for designing multidimensional data structures work for $d = 2$ but not for some higher $d$?