Miloslav Brožek

brozemi1@fit.cvut.cz

presented:

Efficient dynamic range minimum query

Alice Heliou, Martine Léonard, Laurent Mouchard, Mikael Salson

http://www.sciencedirect.com/science/article/pii/S0304397516303164

Definitions

- S is a sequence of integers of length n, $S[i]$ denotes i-th element in the sequence indexed from 0
- Sequence S has a *local minimum* at position i if one of the following holds:

  1. $i = n-1 \wedge S[n-2] > S[n-1]$
  2. $i = 0 \wedge S[0] > S[1]$
  3. $0 < i < n-1 \wedge S[i] \leq S[i+1] \wedge \exists p, 0 < p \leq i : S[p-1] > S[p] \wedge j \in [p..i], S[j] = S[i]$
  4. $0 < i < n-1 \wedge S[i] \leq S[i+1] \wedge \forall j \in [0..i], S[j] = S[i]$

- $S^{[k]}$ is a sequence composed of k-local minima in position order ($S^{[0]} = S$)
- M_k is a bit-vector storing 1 as position i if $S[i]$ element is k-local minima (0 otherwise)
- *plateau* is a maximal interval $[i..j]$ where all values $S[k], k \in [i..j]$ are equal
- P_k is a bit-vector storing 1 at position i if element $S[i]$ is adjacent to exactly 1 equal neighbor – denotes starts and ends of plateaus

Formulas

- **RMQ**

  $$RMQ(S^{[k]}, i_l, i_r) = min(S^{[k]}i_l, S^{[k]}i_r, (*)RMQ(S^{[k+1]}, rank_1(M_{k+1}, i_l), rank_1(M_{k+1}, i_r - 1) - 1))$$

- **Sum estimate**

  $$\sum_{i=1}^{\frac{\log n}{\log^{1+\epsilon}\log n}} \frac{n}{2^{i \log^{1+\epsilon}\log n}} = o\left(\frac{n}{\log n}\right)$$

- **Entropy sum estimate**

  $$H_0(M'_1) + ... + H_0(M'_{k_M}) \leq \sum_{i=1}^{\lceil \log n \rceil + 1} \left(\frac{1}{2^i} \log 2^i + \left(1 - \frac{1}{2^i}\right) \log \left(\frac{1}{1 - \frac{1}{2^i}} \right) \right) < 3.16$$

Navarro Nekrich data structure

- bit-vector with insert/delete and rank/select operations in $O\left(\frac{\log n}{\log \log n}\right)$ amortized time

⇒ *rank(k)* returns number of ones(1) in interval $[0, k]$

⇒ *select(k)* returns position of k-th one(1)

Complexities

Time	**Sample**	**Sparse**
one passage – no update	$O(\log^{1+\epsilon}\log n)$	$O(1)$
total – update	$O(\log^2 n \log^\epsilon \log n)$	$O\left(\frac{\log^2 n}{\log\log n}\right)$

Memory	**Sample**	**Sparse**
no update	$2nH_0(M) + o(n)$	$3.16n + o(n)$
update	$2n(H_0(M) + H_0(P)) + o(n)$	$6.32n + o(n)$