

**Longest Increasing Subsequence Computation over Streaming Sequences**

**Abstract:**

In this paper, we propose a data structure, a quadruple neighbor list (QN-list, for short), to support real time queries of all longest increasing subsequence (LIS) and LIS with constraints over sequential data streams. The QN-List built by our algorithm requires O(w) space, where w is the time window size. The running time for building the initial QN-List takes O(wlogw) time. Applying the QN-List, insertion of the new item takes O(log w) time and deletion of the first item takes O(w) time. To the best of our knowledge, this is the first work to support both LIS enumeration and LIS with constraints computation by using a single uniform data structure for real time sequential data streams. Our method outperforms the state-of-the-art methods in both time and space cost, not only theoretically, but also empirically.

**Existing System:**

Among these, computing the Longest Increasing Subsequence (LIS) over sequential data is a classical problem. An increasing1 subsequence is a subsequence whose elements are sorted in order from the smallest to the biggest. Note that a sequence may contain multiple LIS, all of which have the same length.

Moreover, many works are based on static sequences and techniques developed in these works cannot handle updates which are essential in the context of data streams. To the best of our knowledge, there are only three research articles that addressed the problem of computing LIS over data stream model

**Proposed System:**

We are the first to consider the computation of both LIS with constraints and LIS enumeration in the data stream model.

We introduce a novel data structure to handle both LIS enumeration and computation of all existing LIS with constraints uniformly.

Our data structure is scalable in stream model because of the linear update algorithm and linear space cost. Extensive experiments confirm the superiority of our method.