



Engineering dynamic optimal search algorithm

What:

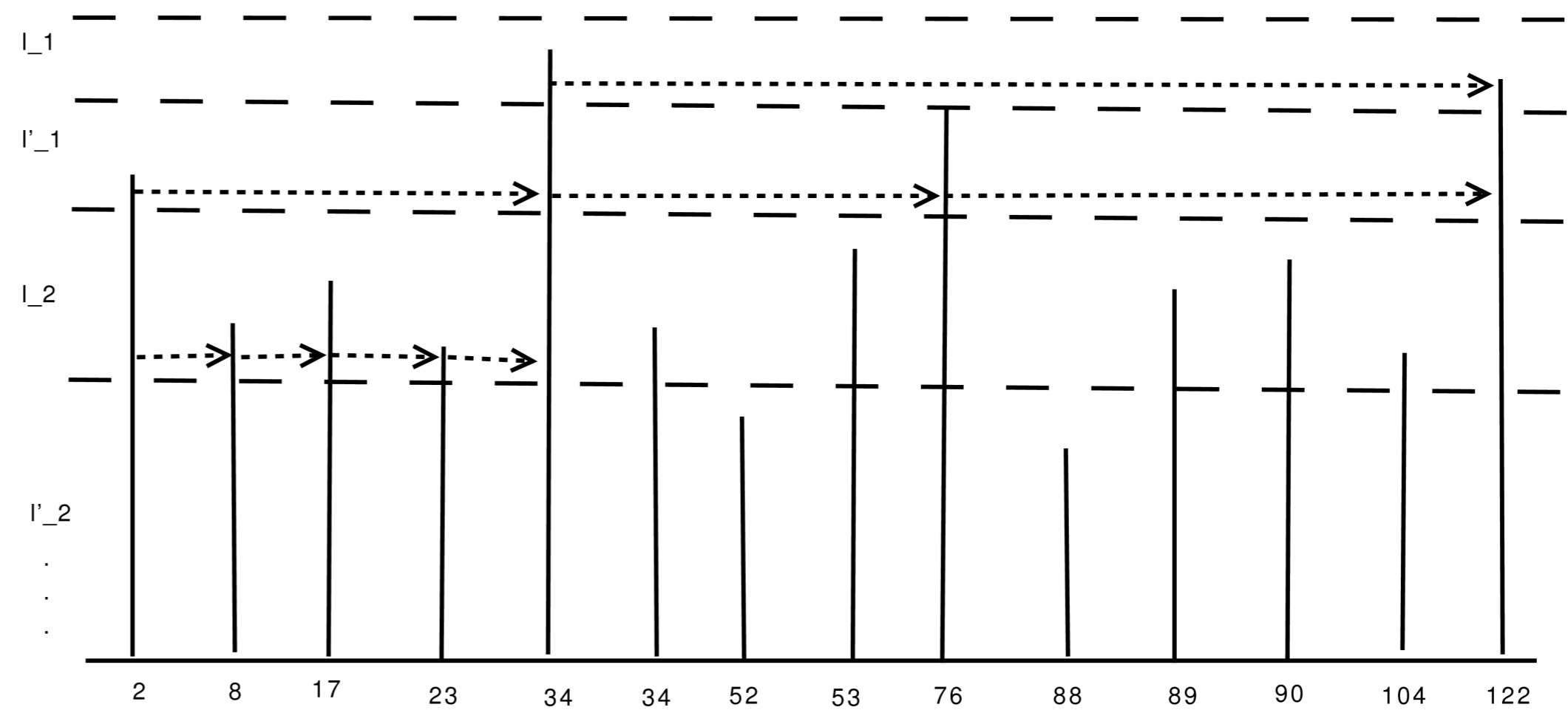
Dynamic problems and in particular dynamic searching is used in many realtime applications. Bose et Al. has developed a dynamic skip list data structure which has an optimal working-set bound both in main memory and in the I/O model.

Problem:

Testing if the skip list developed by Bose et Al. has better performance in internal memory than known algorithms such as the Splay Tree. Comparing standard B-Tree with a dynamic optimal B-Tree for non-uniform searching in external memory and seeing if the external algorithm has better performance.

Solution:

Engineering a dynamic skip list and comparing it to splay tree developed by D. Splayton and R. Tarjan. Also implementing the dynamic optimal B-Tree which is also described in Bose et Al. Article. Which is optimal in the I/O model. In the I/O model it could be tested against normal searching algorithm like B-Tree.



Usage:

Dynamic searching could as an example be used in DNS searching where its common to find domains which is more likely to have requests than the average URL. Other application is of course dictionaries and other (massive) data collections where there isn't a uniform distribution of requested items.

References:

- *Dynamic Optimality for Skip Lists and B-Trees*, P. Bose, K Douieb, S. Langemand
- *Biased Skip Lists*, A. Bagchi, A. L. Buchsbaum, M. T. Goodrich