ENS Lyon. Day 3. Basic group: Problem Analysis

October 28, 2015

Problem A. K-th Maximum

Statement

- Implement the data structure that allows the following queries:
  - Add and remove elements;
  - Find the k-th maximum among elements;

Solution

- Store size field;
- If size of the left subtree is more than k, go to the left;
- Otherwise, go to the right with $k = k - L.size + 1$.

Problem B. Swapper

Statement

- Given an array of n elements;
- Queries of two kinds:
  - Swap adjacent elements in the given range: x with $x+1$, $x+2$ with $x+3$ and so on;
  - Calculate the sum in the given range.

Solution

- Keep two treaps with implicit key: elements with odd and even indices;
- Sum: find the sum in both trees;
- Swap: cut out corresponding ranges in both tree, swap it.

Problem C. Reverse

Statement

- Given an array of n elements;
- Queries: reverse arbitary range, calculate sum in the given range.

Solution

- Treap with implicit key;
- Push: swap left and right subtrees; flip inconsistency in subtrees;
- See lecture notes.

Problem D. Cartesian Tree (Treap)

Statement

- Given n pairs (a_i, b_i);
- Construct cartesian tree.

Solution

- Sort pairs in increasing order of a_i;
- Iteratively add (a_{i+1}, a_{i+1})
  - $b_i > b_{i+1}$, add (a_{i+1}, b_{i+1}) as the right child (a_i, a_i)
  - Otherwise, find parent $v = (a_k, b_k)$ such that $b_k > b_{i+1}$; new node is the right child of v, right child of v is the left child of new node;
  - See lecture notes.

Problem E. Sum

Statement

- Consider set of integers;
- Queries: add element, find sum of all element x s.t. $l \leq x \leq r$.

Solution

- Additional field "sum": sum of elements in subtree with root in this vertex;
- Split the given range, return sum stored in the root;
- Update sum in merge and split operations (in a similar way as size of subtree).

Problem F. Move to front

Statement

- Consider an array of n elements;
- Queries: Move given range in the beginning of the array.

Solution

- Treap with implicit key;
- Move: cut required range;
- Merge trees in the order: range, left part, right part.

Problem G. Key Values

Statement

- Consider array A of infinity length;
- Query $Insert(L, K)$:
  - If $A[i]$, then $A[L] := K$;
  - Otherwise: $Insert(L + 1, A[L])$ and after $A[L] := K$.

Solution

- Treap with implicit key;
- To process "insert": shift suffix of the array to the right.

Problem H. Implicit Key

Statement

- Consider an array of n elements;
- Query: insert element in arbitary position, remove element from arbitary position.

Solution

- Treap with implicit key;
- See lecture notes.

Problem I. Binary Search Tree

Statement

- Implement Binary Search Tree;
- Effective implementation: all queries process at $O(\log n)$ time.

Statement

- Treap;
- Priorities are choosed randomly.

Problem J. Range Minimum Query

Statement

- Consider an array of n elements;
- Queries: insert element in arbitary position, find minimum in the given range.

Solution

- Treap with implicit key;
- Store field for minimum in the subtree; update it in the similar way as size of subtree.

Questions

Questions?