ITMO University
Peking University
Training Camp

ITMO大学北京大学训练营

Day 03: Problem Analysis

Vitaly Aksenov

ITMO University,
St. Petersburg, Russia

25.02.2014

Problems

- Problems, where taken half from the Petrozavodk camp for Russian teams, the other half from the old NEERC Nothern Quaterfinals
- What you should know to solve this problems:
  - Treap (explicit/implicit key)
  - Segment tree
  - Fenwick tree
  - Even in one problem you could use DSU

Problem A. Problem with queries

- You are given some specific subset of sql commands
- And you need to evaluate them

Problem A. Problem with queries

- You should see, that restrictions in this problem are small. This means, that you shouldn’t think to much about optimization of problem and you can use strict-forward solution.
- You need to parse input commands. It seems to be one of the hardest part of the problem, because you need to write a lot of code without mistakes. Such as InSErT is the same as INSERT.
- You need to write the logical operations properly, where AND has bigger priority.

Problem B. Examinator

- You need perform the following queries:
  - How many cubes between A and B?
  - Insert cube A upon cube B.
  - Remove cube A.

Problem B. Examinator

- Let us think, that you don’t need to remove cubes. We will perform adding cubes to the tower with the help of the list structure.
- When we add all the cubes, we know the position of each cube in the result tower. A -> pos[A]
- Now, rephrase all the queries:
  - How many ones between pos[A] and pos[B]?
  - Put one in the position pos[A].
  - Remove one from the position pos[A].
- Now the problem becomes the standard problem on tree.

Problem C. Range Minimum Query

- You need to perform the following queries:
  - Return the minimal element between the i-th and j-th element of the list
  - Add element x after the i-th element

Problem C. Range Minimum Query

- This is the standard problem on the treap with implicit key.
- In the node we will hold not only the sum in the subtree, but the minimum in the subtree. It is easy to recalculate that two values, when you split or merge this nodes.

Problem D. Quadtree

- You are given colored quadtree
- You need to calculate the maximum unicolored area

Problem D. Quadtree

- The easiest solution is to emulate all the division of square recursively.
- The recursion will return 4 vectors of squares for the upper, left, bottom and right borders.
- On each step we firstly go recursively to 4 squares if needed. Afterwards we union borders, for example with dsu.

Problem D. Quadtree

- The second solution, which is quite straightforward.
- We calculate the position of each square, after we go with swipe line and build a graph. Because the graph is planar, the amount of edges isn’t bigger that 3V. After we run dfs, which find the connected components.

Problem E. Key Insertion

- You need to implement following query:
  - Insert the value K in the first empty position, which is bigger or equal than L

Problem E. Key Insertion

- This problem can be solved with treap.
- Each node of treap contains left and right ends of continuous part of the array.
- When you insert in position L:
  - you can get into the existing segment, and you insert K into the inner treap in this node.
  - you create create additional node in treap for [L, L].
- You have 3 possibilities afterwards:
  - Union this node with nothing
  - Union this node with the next one
  - Union this node with the next and the previous

Problem F. K-th Number

- You should answer on the query: What would be the k-th number in a[i..j] segment.

Problem F. K-th Number

- $O(\log^2 n)$ per query. Offline.
- We run binary search on every segment at the same time.
- We get the following queries: give me the amount of number, which are less, than k in the [l, r].
- This can be easily done in O(n logn) time.

Problem F. K-th Number

- $O(\log^2 n)$ per query. On-line.
- We build the merge-sort segment tree.
- We take the O(logn) segments, which are represent [l, r].
- Run the binary-search on the answer.

Problem F. K-th Number

- O(log n) per query and O(n log n) memory. On-line.
- It can be done with persistent segment tree for each prefix. When you need to evaluate [l, r] query – you take *tree*[l-1] and *tree*[r].

Problem G. Drying

- You are given n clothes.
- You can choose one thing to decrease by k at a time.
- All others will decrease by 1 at a time.

Problem G. Drying

- Let us run binary search on the amount of time, which we will spend on drying.
- Let is be t. We will decrease each a_i on t. And we can decrease some numbers at *k*-1 at total t times.

Problem H. Inner Vertices

- There are n points on the board.
- In each step the white point, which has black points in the upper, left, bottom and right directions becomes black.
- The total amount of black points after all steps have been done.

Problem H. Inner Vertices

- The first idea should be, that you have only one step at all time. And all black vertices after step are situated on the cross of vertical and horizontal segments, which ends are black.
- To count the number of intersections, we will use the standard technique of swipe line and events with segment tree.

Problem I. Jenny's First Exam

- You are given n jobs, which you need to be done between t_1 and t_2.
- And you should start doing jobs as later as possible.

Problem I. Jenny's First Exam

- If there wasn't any restrictions on the start time, it will be the standard Job-shop problem.
- The key idea is to revert all the jobs in time, the end becomes the start, and the start becomes the end.
- After that we are going on ends in decreasing order. And in every day we try to do the job, which has the bigger start time.

Problem J. K Best

- You are given n jewels with 2 characteristics v and w.
- You need to choose k of them, which maximize the following ratio: $\frac{\sum_{j=1}^{k} v_{i_j}}{\sum_{j=1}^{k} w_{i_j}}$.

Problem J. K Best

- Let s’ be some fixed value and s maximal possible specific value
  - What is ration between s and s’?
- As $s(S) = \frac{\sum_{j=1}^{k} v_{ij}}{\sum_{j=1}^{k} w_{ij}}$, so $\sum_{j=1}^{k} \left(v_{ij} - w_{ij} s(S) \right) = 0$
- Compute this value for s’ and compare with zero
  - If it greater than 0, then $s' < s(S)$
  - If it less than $s' > s(S)$

Problem J. K Best

- Binary search for s’
- For fix value s’
  - Sort all jewels by $v - s'w$
  - Take k first
  - Compute $\sum_{j=1}^{k}(v_{i_j} - s'w_{i_j})$

Thank you!

Thank you for your attention!