Update submatrix and query submatrix.

If we are given a matrix of NxN (N<=1000). There are 2 operations:

4

1. update x1, y1, x2, y2, val. In this operation, val is added to each element in the sub-matrix (x1, y1, x2 y2).
2. Query x1, y1, x2, y2. In this operation, the sum of all the elements in the sub-matrix (x1, y1, x2, y2).

How do I perform them efficiently? I tried coding some sort of a 2-d segment tree but that doesn't work. Quadtree cant be used as it is O(N) in worst case.

1

segment-tree algorithm

edited **08 Sep, 05:17**
michal27
730●1●6●15

asked **08 Sep, 05:01**
pushkarmishra
316●7●15●23
accept rate: 0%

One Answer:

Very nice question. I spend a while thinking about it. And I believe, that 2-d segment tree is a right way to solution. So my approach.

2

First, let's focus on easier problem. The main issue is update, so suppose, we only want to update one element at the time - to element on position *x*, *y* add value *val*. The second query remain the same.

Let *n* be equal to some power of 2. Now we can use 2-d segment tree. That is construct as follows. First build one segment tree on rows (we will call it "upper" segment tree). Each node of this upper tree corespond to some interval of rows. And the node can be represented as one row of length *n*, which is sum of respective rows. Now in each node build one "bottom" segment tree - the segment tree on columns in respective rows (so on row created as sum of them). This is an ordinary segment tree.

Whit this, we can easily perform queries and updates. Just do segment search on upper tree with rows and when we reach final node (when we don't want to nested to sons) we switch do bottom tree of this node and search segment of

columns. Update is perform similary but in reverse order - just like in classic segment tree. The time complexity of this is *O(log^2 n)* because in *log n* nodes of upper tree we must go to *log n* nodes in bottom tree. And this can be even implemented very easily, if we use Fenwick tree. Here is my code for 2-d Fenwick tree.

Now try to solve original problem, when we want to update whole rectangle. When we have segment tree, we use lazy-loading approach and this should work here as well. Lazy flag in some node of segment tree means, that interval corresponding to this node must be change by this value, but we don't need to do it right now. Next time, when we visited this node and we need correct value, we update values. This save us lots of time and we do updates only, when they are necessary.

But we have two levels of trees, so we need two type of lazy flags. In bottom trees it's just ordinary lazy flag - I must update value in my segment by *this* value. In upper tree we use different flag with meaning - in segment of my rows I must update *this* segment of columns by *this* value. *This* is some variable, which that node remember.

Again this should be *O(log^2 n)*. **But** I'm not sure yet, if this solution is absolutely correct. I can overlook some corner cases, but I believe that, this should be main scheme to 'how to solve this problem'. And also the implementation of this can be really tricky and awfull. So good luck, I hope I help you with this.

answered **08 Sep, 17:20**
michal27
730●1●6●15
accept rate: 13%

Unfortunately, this will work in O(M^2) for M queries in worst case.

Each node of upper tree can have multiple lazy tags, so to push all tags down we will have to perform O(M) operations in worst case, for example, 2N updates:

First N updates to range [1..N][L..R], where L and R are different for each update. And second N - to range [P..P][1..1], for each 1 <= P <= N. So, after the first group of updates root of upper tree will have N lazy tags and after second group each of them will be pushed to leaves through each of 2N nodes, so in total 2N^2 pushes will be performed.

alex_2oo8 (08 Sep, 21:37)

yes, I see, you're right :(

Tags:

algorithm **×114**
segment-tree **×20**

Asked: **08 Sep, 05:01**
Seen: **275 times**
Last updated: **08 Sep, 21:42**

Related questions

- What is the difference between Binary Indexed Tree and Segment Tree? Please explain.
- Longest Increasing Subsequence
- Floyd Warshall Algorithm
- please Explain how to solve RDF
- Yandex.Algorithm 2013 in Saint Petersburg, Russia, $18000 in prizes
- spoj question
- Sudoku Generate | Randomized Algorithm
- Generate all the k-element subsets from a n-element set (n>k).
- Please make the solutions of Flipping coins (http://www.codechef.com/problems/FLIPCOIN/) public.
- Fitting Lines into a Grid Matrix (Fitting Problem)

yes, I see, you're right :(

michal27 (08 Sep, 21:42)

add new comment

Your answer

[hide preview] community wiki

Post Your Answer