CS573 Course Project Proposal

Ke Chen

March 30, 2006

Question: Let $A[1 \ldots n][1 \ldots n]$ be a $n \times n$ matrix consisting of integers. We have the following operations:

- addcolumn$(i; j, k)$ adds 1 or -1 to elements $A[i][j], A[i][j+1], \ldots, A[i][k]$, then return a maximal item $\max_{x,y} A[x][y]$ of A.
- addrow$(i; j, k)$ adds 1 or -1 to elements $A[j][i], A[j+1][i], \ldots, A[k][i]$, and then return a maximal item $\max_{x,y} A[x][y]$ of A.

Suppose that $A = 0$ initially. The question asks us to design a data structure to maintain A under the above operations such that the amortized cost of each operation is small. It is not hard to support addcolumn$(i; j, k)$ and addrow$(i; j, k)$ in $O(n)$ time. Unfortunately, we do not know any (significantly) better algorithm than this. We are interested in a data structure to support addcolumn$(i; j, k)$ and addrow$(i; j, k)$ in, say, $O(\sqrt{n})$ time. (Actually, any $o(n)$ time algorithm may be interesting.)

We refer to the problem as `DSProb`.

Motivation: The motivation comes from a geometric optimization problem: Given a set P of points in the plane, a rectangle Q, and a parameter k, if *translation* and *rotation* are allowed, can Q cover at least k points of P? We refer to this geometric problem as `GeomProb`.

`GeomProb` can be reduced to `DSProb`; in particular, if the amortized cost of each operation of `DSProb` is $f(n)$, then we can solve `GeomProb` in $O(n^2 \cdot f(n))$ time (it is easy to show an $O(n^3 \log n)$ time algorithm for `GeomProb`). The reduction is a little bit tedious to describe, and is omitted here.

Other than the application mentioned above, `DSProb` may find applications in other problems. Also, the problem itself seems quite intriguing.

Related Work: In `DSProb`, if only the operation of addcolumn$(i; j, k)$ (or addrow$(i; j, k)$) is supported, then it is not hard to adapt known data structures (e.g., 1d-trees, essentially complete binary search trees) to solving this problem. For a description of kd-trees, see Chapter 5 of "Computational geometry: algorithms and applications" [dBvKOS00]. There is some hope that one can adapt 2d-trees or range trees (see Chapter 5 of [dBvKOS00]) to solving `DSprob`, but we do not know how. Another related question is, if one always add 1 in the operations addcolumn$(i; j, k)$ and addrow$(i; j, k)$, does the problem become easier? Also, can one lower bound the cost of each operation in `DSProb`?

For `GeomProb`, Erickson, Har-Peled, and Mount [EHM05] considered the following related problem: Given a set P of points in the plane, how to compute the narrowest slab bounded by two parallel lines that contains k points of P? (They also considered higher dimensional cases.) For the problem considered, they presented exact and approximation algorithms and provided nearly matching lower bounds under certain assumptions.

Credits: `GeomProb` was brought up by Feida Zhu. Ke Chen, Jeff Erickson, Sariel Har-Peled, and Feida Zhu contributed to the reduction from `GeomProb` to `DSProb`.

References

[dBvKOS00] M. de Berg, M. van Kreveld, M. H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2nd edition, 2000.

[EHM05] J. Erickson, S. Har-Peled, and D. Mount. On the least median square problem. *Discrete Comput. Geom.*, 2005. To appear.