Tight Lower Bound for the Partial-Sums Problem

Ke Yi

April 19, 2005

In the partial-sums problem, we are given an array $A[1, \ldots, n]$ and want to support the following two operations:

- `update`(k, Δ): $A[k] \leftarrow A[k] + \Delta$;
- `sum`(i, j): return $\sum_{k=i}^{j} A[k]$.

This is perhaps one of the most fundamental data structure problems. When the values of $A[1, \ldots, n]$ are drawn from a semigroup, its static version (where updates are not allowed) has the famous inverse-Ackermann $\Theta(\alpha(n))$ query bound with linear storage [4]; the dynamic version can be easily solved by a binary tree in $O(\log n)$ time per operation, and a corresponding $\Omega(\log n)$ lower bound was proven in [2]. In the group model, where we are allowed to do subtractions, the static version can be trivially solved in constant time per query by storing the prefix sums. However, people have not been able to use the power of subtractions to beat the simple binary tree solution for the dynamic case, and an $\Omega(\log n/ \log\log n)$ lower bound was established in [1]. In 2004, an MIT freshman and Erik Demaine broke this record and proved a matching $\Omega(\log n)$ lower bound for this dynamic partial-sums problem in the group model [3]. Their technique is very novel, quite clean, and requires a minimal combinatorial effort. This lower bound also holds for the offline case and can be easily extended to obtain a tradeoff between query and update times. I will present their basic lower bound proof and briefly sketch the extensions.

References

[1] M. L. Fredman and M. E. Saks. The cell probe complexity of dynamic data structures. In *Proc. 21st Annu. ACM Sympos. Theory Comput.*, pages 345–354, 1989.

[2] H. Hampapuram and M. L. Fredman. Optimal biweighted binary trees and the complexity of maintaining partial sums. *SIAM J. Comput.*, 28:1–9, 1998.

[3] M. Pătraşcu and E. D. Demaine. Tight bounds for the partial-sums problem. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms*, pages 20–29, 2004.

[4] A. C. Yao. Space-time trade-off for answering range queries. In *Proc. 14th Annu. ACM Sympos. Theory Comput.*, pages 128–136, 1982.