Problem set for the Algoritmica 2 class (2016/7)

Roberto Grossi
Dipartimento di Informatica, Università di Pisa
grossi@di.unipi.it

October 5, 2016

Abstract

This is the problem set assigned during class. What is relevant during the resolution of the problems is the reasoning path that leads to their solutions, thus offering the opportunity to learn from mistakes. This is why they are discussed by students in groups, one class per week, under the supervision of the teacher to guide the brainstorming process behind the solutions. The *wrong* way to use this problem set: accumulate the problems and start solving them alone, a couple of weeks before the exam. The correct way: solve them each week in groups, discussing them with classmates and teacher.

1. [Range updates] Consider an array C of n integers, initially all equal to zero. We want to support the following operations:

   - `update`(i, j, c), where $0 \leq i \leq j \leq n-1$ and c is an integer: it changes C such that $C[k] := C[k] + c$ for every $i \leq k \leq j$.
   - `query`(i), where $0 \leq i \leq n-1$: it returns the valus of $C[i]$.
   - `sum`(i, j), where $0 \leq i \leq j \leq n-1$: it returns $\sum_{k=i}^{j} C[k]$.

   Design a data structure that uses $O(n)$ space and implements each operation above in $O(\log n)$ time. Note that `query`(i) = `sum`(i, i) but it helps to reason. [Hint to further save space: use an implicit tree such as the Fenwick tree (see wikipedia).]