ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SS 2006

Prof. R. Wattenhofer / Prof. P. Widmayer / T. Locher / R. Flury

Principles of Distributed Computing
Exercise 9

1 Segmented Prefix Sums

We are given a sequence $A = (a_1, a_2, \dots, a_n)$ of elements from a set S with an associative operation $*$, and a Boolean array B of length n such that $b_1 = b_n = 1$. For each $i_1 < i_2$ such that $b_{i_1} = b_{i_2} = 1$ and $b_j = 0$ for all $i_1 < j < i_2$, we wish to compute the prefix sums of the subarray $(a_{i_1+1}, \dots, a_{i_2})$ of A. Develop an $O(\log n)$ time algorithm to compute all the corresponding prefix sums. Your algorithm should use $O(n)$ operations and should run on the EREW PRAM.
The results are written into an array r (see Figure 1 for a numeric example).

a	**25**	**27**	**1**	**7**	**31**	**67**	**12**	**89**	**14**	**55**
b	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**
r	**35**	**0**	**0**	**98**	**0**	**12**	**158**	**0**	**0**	**0**

Figure 1: The prefix sums are written into the array r. There is a 0 at a given index in r, if the corresponding entry of b is also 0. The last element of r is always 0, as there are no subsequent elements in b.

2 Prefix and Suffix Minima

Let $A = (a_1, a_2, \dots, a_n)$ be an array of elements drawn from a linearly ordered set. The *suffix minima problem* is to compute for each i, where $1 \le i \le n$, the minimum element among $\{a_i, a_{i+1}, \dots, a_n\}$. We can, in a similar fashion, define the *prefix minima*. Develop an $O(\log n)$ time algorithm to compute the prefix and the suffix minima of A using a total of $O(n)$ operations. Your algorithm should run on the EREW PRAM.