Principles of Distributed Computing
Exercise 9: Sample Solution

1 Segmented Prefix Sums

The result can be obtained in the following manner. We use the algorithm presented in the lecture to compute the prefix sums of array $a$ and write the results into a helper array $h_1$. We also compute the prefix sums of array $b$ and write the results into a second helper array $h_2$. Clearly, both computations require $O(\log n)$ time and $O(n)$ operations.

In the next step, each processor $p_i$ where $b[i] = 1$ writes $i$ into a third helper array $h_3$ at position $h_2[i]$. Since the values in $h_2$ differ for all processors $p_i$ where $b[i] = 1$, there is no concurrent access to any cell in array $h_3$. Subsequently, all processors $p_i$ where $b[i] = 1$ read the value $h_3[h_2[i] + 1]$, i.e. the next higher index with a 1 in array $b$. Then, each processor $p_i$ where $b[i] = 1$ reads the value at position $h_3[h_2[i] + 1]$ in array $h_1$, which is the prefix sum up to this index.

In a final step, the processors $p_i$ where $b[i] = 1$ subtract $h_3[i]$, i.e. the prefix sum up to its own index, from $h_1[h_3[h_2[i] + 1]]$, i.e. the prefix sum up to the next index with a 1 in array $b$, and write this value into the array $r$ at position $i$. This concludes the computation.

These additional steps after the computation of array $h_1$ and $h_2$ can be performed in $O(1)$ time using $O(n)$ operations. Thus, in total the computation requires $O(\log n)$ time and $O(n)$ operations. There are no parallel accesses to the same cells in any array, thus the algorithm runs on the EREW PRAM.

2 Prefix and Suffix Minima

In order to compute the prefix minima, we can use the same algorithm as for the computation of the prefix sums, only the addition has to be replaced by a min operation.

The suffix minima can be computed by first reversing the array in constant time and applying the prefix sum algorithm, again with the min operation, to the reversed array.

---

1 All the processors $p_i$ where $b[i] = 0$ simply write a 0 into the array $r$. 