Write for the solution of Exercise 2 and 3 the number of used Processors, the size of the memory, the used PRAM model and the running time of your algorithm.

**Exercise 1** (4 points)

(a) Give a hierarchy for the PRAM models (CREW, CRCW, EREW) and the writing rules (Common, Arbitrary, Priority) and explain it.

(b) Prove that a Priority-CRCW PRAM with \( p \) processors can be replaced by a EREW PRAM with \( p \) processors such that the running time increases by a factor of \( O(\log(p)) \).

Hint: Give a method to solve read conflicts and one to solve write conflicts.

**Exercise 2** (6 points)

Assume you have an array \( A \) with \( n \) entries which are 0 or 1. Give a CRCW PRAM algorithm in Common or Arbitrary mode (not Priority) that calculates the smallest index \( k \) such that \( A[k] = 1 \) holds (The first position where the entry of the array is 1). Your algorithm should achieve a running time of \( O(1) \) while using not more than \( O(n) \) processors.

Hint: Think about separating the array into blocks of size \( \sqrt{n} \).

**Exercise 3** (4 points)

Assume you have \( n \) intervals \( (x_i, y_i) \) with \( x_i < y_i \) for \( 1 \leq i \leq n \). The \( 2n \) endpoints are ordered from smallest to highest and if two points have the same value, the one with lower index comes first. Assume the input has \( 4n \) memory blocks. On every odd memory block, an endpoint of the ordered intervals is written such that the smallest endpoint is written in the first block. The index of the interval is written into the next memory block, such that on every even position \( i \) the index of the endpoint of the memory block \( i - 1 \) is written.

Design a PRAM algorithm that assigns each interval a horizontal line, such that two intervals that are assigned to the same line do not cross each other and that the number of horizontal lines is minimized. Your algorithm should have a running time of \( O(\log(n)) \).