B – Simple merge

in4026

For the list of requirements of this lab course exercise, please read the lab course manual (can be downloaded from the BlackBoard), especially Section 1.

Let $X = (x_1, \ldots x_n)$ be a sequence of elements drawn from a linear ordered set S. The *rank on an element* $x \in S$ in X, denoted by rank$(x : X)$ is the number of elements of X smaller than or equal to x. If $Y = (y_1, \ldots, y_M)$ is another sequence with elements from S (i.e., $Y \neq X$), then rank$(Y : X) = (r_1, \ldots, r_M)$, where $r_i =$ rank$(y_i : X)$ is the rank of Y in X.

Merging two arrays A and B is equal to determining rank$(A : AB)$ and rank$(B : AB)$, where AB is the concatenation of A and B.

Design and implement an efficient parallel program in PThreads and OpenMP that merges two sorted sequences using the method above. We recommend the use of an algorithm with time complexity $\mathcal{O}(\log(N + M))$, however, any solution with complexity not worse than $\mathcal{O}(N + M)$ is acceptable if explained why it is preferred.

Test your program, among others, with the following two input vectors

A =	28	36	40	61	68	71	123	149
B =	2	5	18	21	24	29	31	33
	34	35	47	48	49	50	52	62
	66	70	73	80	88	89	114	124
	125	131	143	144	145	148	155	159

Note that normal I/O (e.g., reading the input array from a file, displaying the solution) is not considered part of the algorithm's time complexity.