Problem 16

(6 points) Design a parallel algorithms that merges two sorted arrays into one sorted array in time O(1)

using a polynomial number of processors on a CRCW Common PRAM. Let our two sorted input lists, A and B, be of lengths m and n respectively. Our algorithm will run on (m + n)2 processors. To each (m + n) location in memory, assign (m + n) processors. Consider the ith value of the input in, say, B. These processors are given the task of determining how many values in A and B are of greater value than their A[i]. There are (m + n 1) locations to consider, and (m + n) so this can be done in constant time by appending/summing to a specified location in memory, say S[i] (utilizing our CRCW Common hardware). The remaining processor will right A[i] after all processors have completed. We can determine this by appending to a common location in memory (simply add 1 to a specified location for the processor; the processors will be done when its value is m + n 1).

Now, once our final processor sees that the (m + n 1) processors in its group have completed, it writes A[i] to the position p=S[i] in the output.