

Sorting Networks and Merge Sort

Name: _____ Student Number: _____

Name: _____ Student Number: _____

Name: _____ Student Number: _____

1. **Do you remember the good old sequential days?** Assume you are given two sequences A and B of length N that are already sorted. Answer the following questions using “big-O” notation and assuming worst-case.

(a) How many comparisons does it take to merge the sequences?

(b) How much time does it take to merge the sequences using a single thread?

2. **Merging using sorting networks: Warmup.** See the other side of the page.

3. **Merging using sorting networks: General case.** Choose two sorted sequences of four random numbers each (for convenience, we recommend numbers between 0 and 99 inclusive) and write them in the A and B sequences on the left side of the sorting worksheet. Fill in the rest of the worksheet and see if the network sorts your numbers into sequence C . The white/grey color backgrounds on the boxes are designed to help you with the data shuffle between stages.

In the questions below, give an answer both for the specific sequences on your worksheet (where $N = 4$) and for general N . Give a specific number in the first case and use “big-O” notation for the second.

(a) How many stages involving compare and swap modules are necessary?

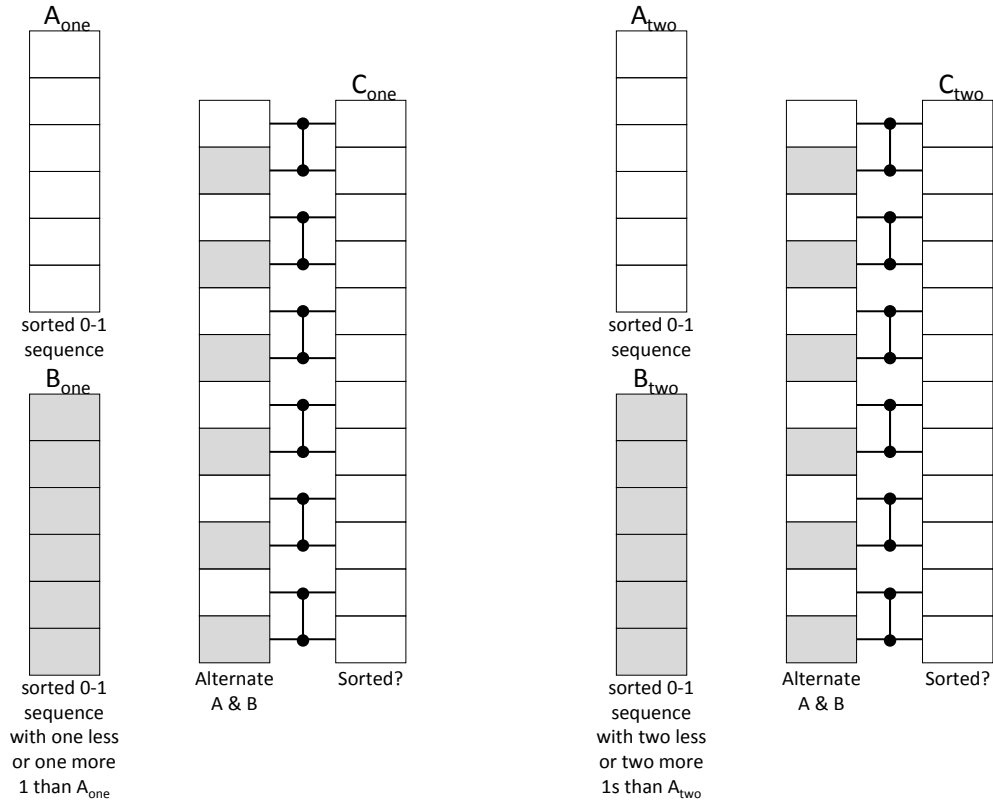
(b) How many compare and swap modules does it take at each of the stages in which such modules appear?

(c) How many compare and swap modules does it take in total?

(d) How much time does it take to perform the merge at each of the stages in which such merges occur?

(e) How much time does it take to perform the tree of merges, assuming that the shuffle between stages takes zero time?

Fill in the A and B sequences as directed in the two examples below, and then use the simple sorting network to generate the merged C sequence for each case.



- How many compare and swap modules does it take to merge in each of the two cases?
- How much time does it take to merge in each of the two cases?