Lab02-Divide and Conquer

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Jiahao Fan.

∗ Name:________ Student ID:________ Email: ___________

1. Consider the following recurrence:

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ 2T(n/2) + O(n \log n) & \text{if } n = 2^k \text{ and } k \geq 1 \end{cases}$$

   (a) Solve $T(n)$ (in the form of O-notation) by recurrence tree. Detailed derivation is required.

   (b) Can we use the Master Theorem to solve this recurrence? Please explain your answer.

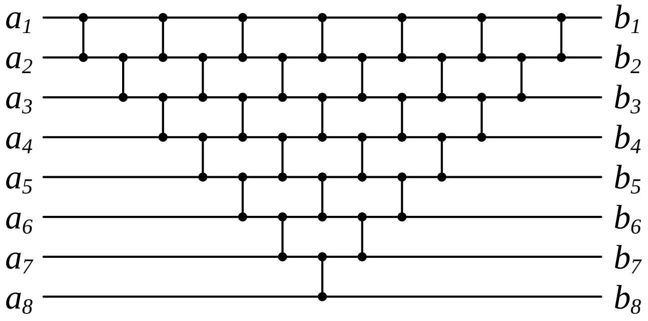
2. Given any array num, find the number of pairs (i, j) satisfying $i < j$ and $num[i] > 2 \times num[j]$. For example, if $num = [1, 3, 2, 3, 1]$, the answer should be 2.

   (a) Design an algorithm to solve this problem using divide-and-conquer strategy and complete the implementation in the provided C/C++ source code. (The source code *Code-Pairs.cpp* is attached on the course webpage.)

   (b) Write a recurrence for the running time of your algorithm and solve it using the Master Theorem directly.

3. **Transposition Sorting Network**: A comparison network is a **transposition network** if each comparator connects adjacent lines, as in the network in Fig. 1.

Figure 1: A



   (a) Prove that a transposition network with n inputs is a sorting network if and only if it sorts the sequence $\langle n, n-1, \cdots, 1\rangle$. (Hint: Use an induction argument analogous to the *Domain Conversion Lemma*.)

   (b) (Bonus) Given any $n \in \mathbb{N}$, write a program using Tkinter in Python to draw a figure similar to Fig. 1 with n input wires.

Remark: include your .cpp, .py, .pdf and .tex files in your uploaded .rar or .zip file.