CS 141: Advanced Topics in Algorithms  
Spring 2006  
Sorting Networks Homework

Due: Wednesday, March 3 by 5 PM under Ran’s Office Door

1. [10 Points] **Odd-Even Mergesort!** Recall the Odd-Even Mergesort algorithm from class.

**Odd-Even Merge:**

- Input: a sequence $a_1a_2\ldots a_n$, where both halves ($a_1a_2\ldots a_{n/2}$ and $a_{n/2+1}a_{n/2+2}\ldots a_n$) are sorted.
- Step 1: Recursively merge the even-indexed subsequence $a_2a_4a_6\ldots a_n$.
- Step 2: Recursively merge the odd-indexed subsequence $a_1a_3a_5\ldots a_{n-1}$.
- Step 3: For all $i \in \{1,3,5,\ldots n-3\}$, compare $a_i$ and $a_{i+1}$ with a comparator (so the two will be swapped if $a_i > a_{i+1}$).

**Odd-Even Mergesort:** Given a sequence $a_1a_2\ldots a_n$, recursively sort both halves ($a_1a_2\ldots a_{n/2}$ and $a_{n/2+1}a_{n/2+2}\ldots a_n$). Then merge them using Odd-Even Merge.

Prove the correctness of **Odd-Even Mergesort** and show that its depth is $O(\log^2 n)$.  
You may use without proof the result from class that **Odd-Even Merge** has a depth of $O(\log n)$.