Batcher Bitonic sorting algorithm

Basic concepts

• A sequence $a_0, \ldots, a_{n-1}$ is called Bitonic if there is an element $a_i$, $0 < i < n-1$ such that one of the following is satisfied:
  1) $a_0 \leq a_1 \leq \ldots \leq a_i \geq a_{i+1} \geq \ldots \geq a_{n-1}$ or
  2) $a_0 \geq a_1 \geq \ldots \geq a_i \geq a_{i+1} \leq \ldots \leq a_{n-1}$ or
  3) An index shift will satisfy any of the above two relations

Note: condition 2 is not needed. It can be obtained from 1 and 3.
**Theorem:** Given a bitonic sequence \( a_0, \ldots, a_{2n-1} \), let

\[
x_i = \min\{ a_i, a_{i+n} \} \quad \text{for } i=0,\ldots,n-1
\]

\[
y_i = \max\{ a_i, a_{i+n} \} \quad \text{for } i=0,\ldots,n-1
\]

Then each of \( x_0, \ldots, x_{n-1} \) and \( y_0, \ldots, y_{n-1} \) are Bitonic sequences and each element in the first sequence is smaller than any element in the second sequence.

**Sorting a Bitonic sequence**

- Given a n-element bitonic sequence, apply the theorem recursively

- After \( \log n - 1 \) steps, each Bitonic sequence will have only two elements. Which can be trivially sorted.
**Bitonic sorting of** \( n \) **elements**

1) Sort the first \( n/2 \) elements in ascending order and the last \( n/2 \) elements in descending order.

2) Sort the resulting Bitonic sequence in \( \log n \) steps.

---

**Example: sorting 32 elements**

1) Sort sixteen 2-elements bitonic sequences in alternating ascending and descending orders (in one step)

2) Then sort the eight 4-elements bitonic sequences in alternating polarity

To obtain four 8-elements bitonic sequences (in 2 steps).
3) Then sort four 8-elements bitonic sequences

Sort in ascending order | Sort in descending order | Sort in ascending order | Sort in descending order

To obtain two 16-elements bitonic sequences (in 3 steps).

4) Then sort two 16-elements bitonic sequences

Sort in ascending order | Sort in descending order

To obtain one 32-elements bitonic sequences (in 4 steps).
5) Then sort the 32 elements bitonic sequences (in 5 steps).

**Computation of execution time:**

Log $n$ steps,
Each step, $i$, requires $i$ sub-steps. Hence

$$\text{number of steps} = \sum_{i=1}^{\log n} i = \frac{1 + \log n}{2} \log n$$