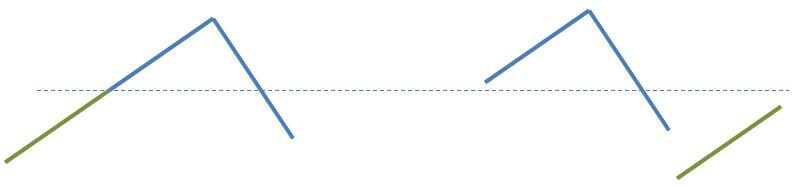
## Batcher Bitonic sorting algorithm

## Basic concepts

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



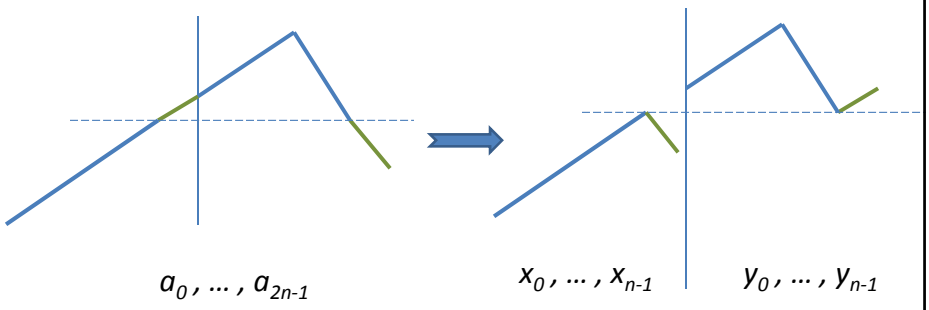
Note: condition 2 is not needed. It can be obtained from 1 and 3.

**Theorem**: Given a bitonic sequence $a_0, \ldots, a_{2n-1}$, let

$x_i = min\{ a_i, a_{i+n} \}$ for $i=0,\ldots,n-1$

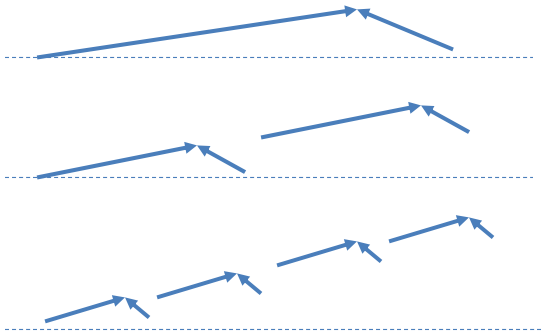
$y_i = max\{ a_i, a_{i+n} \}$ for $i=0,\ldots,n-1$

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

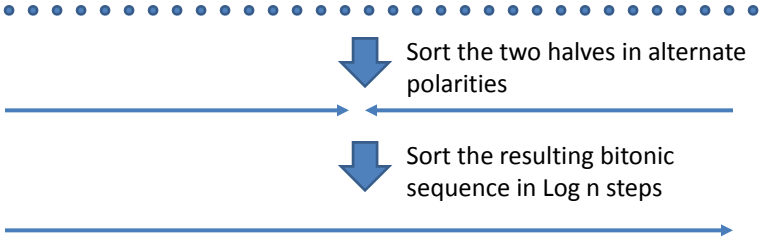
- Given a n-element bitonic sequence, apply the theorem recursively



- After log n - 1 steps, each Bitonic sequence will have only two elements. Which can be trivially sorted.

## Bitonic sorting of n elements

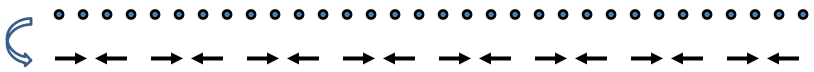
1) Sort the first n/2 elements in ascending order and the last n/2 elements in descending order.
2) Sort the resulting Bitonic sequence in log n steps.



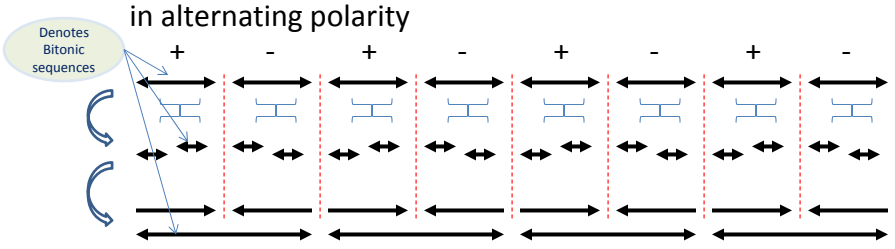

How do you sort n/2 elements? → Recursively

## Example: sorting 32 elements

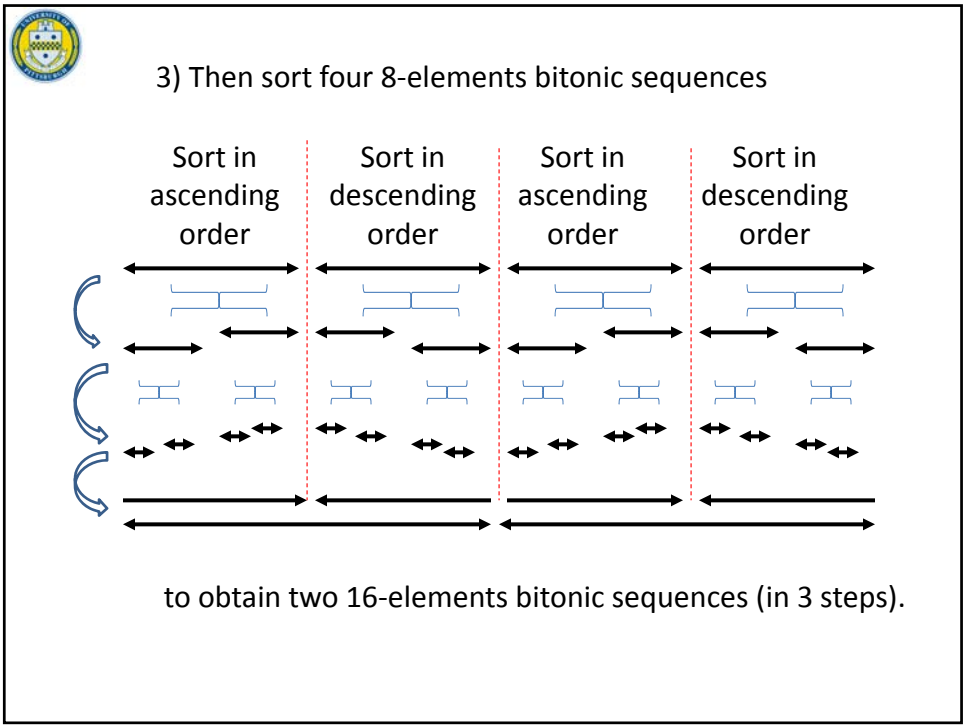
1) Sort sixteen 2-elements bitonic sequences in alternating ascending and descending orders (in one step)



to obtain eight 4-elements bitonic sequences.
2) then sort the eight 4-elements bimodal sequences in alternating polarity



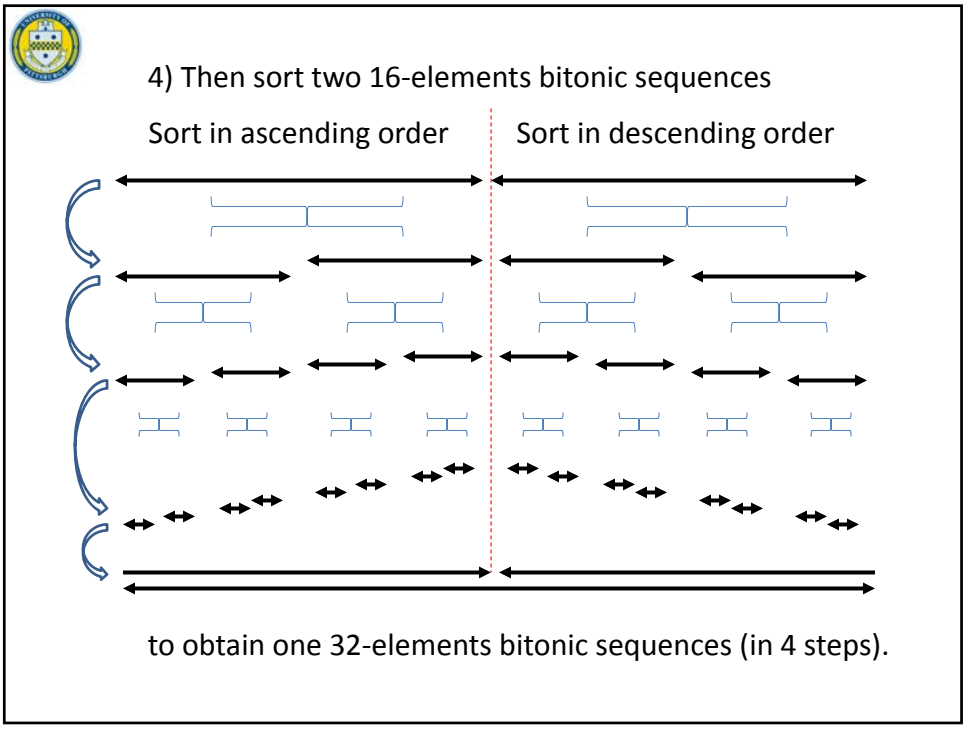

to obtain four 8-elements bitonic sequences (in 2 steps).



3) Then sort four 8-elements bitonic sequences



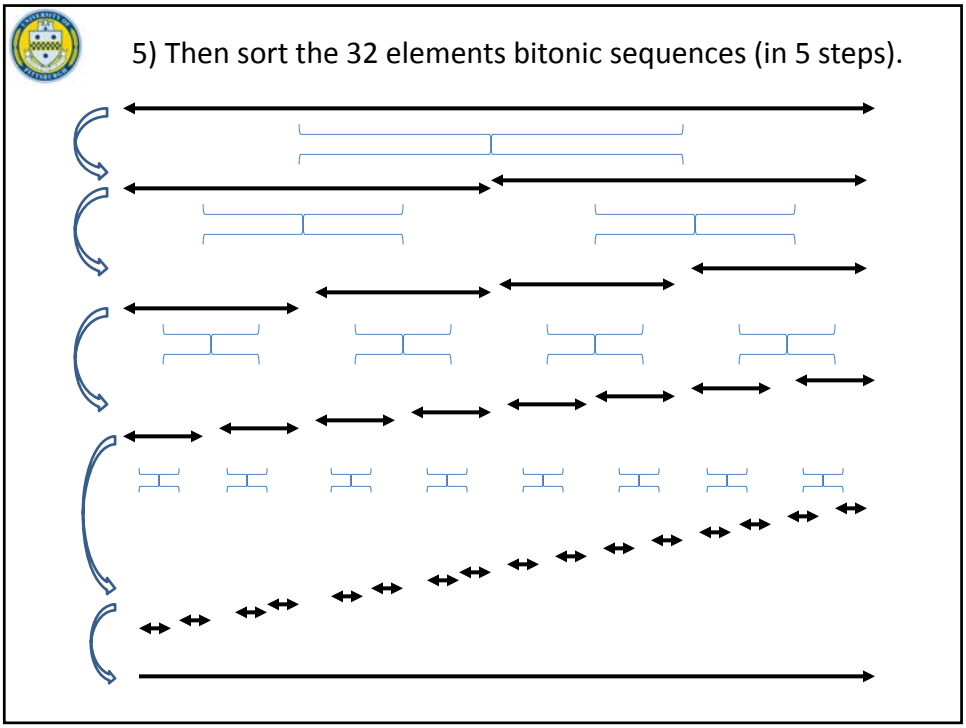
to obtain two 16-elements bitonic sequences (in 3 steps).



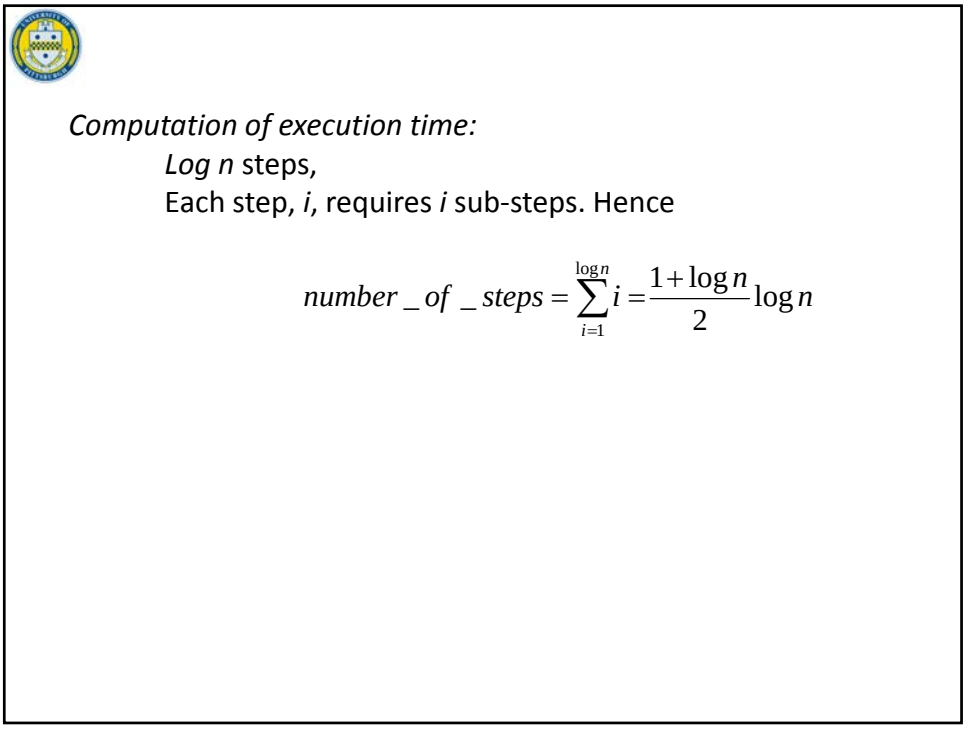
4) Then sort two 16-elements bitonic sequences



to obtain one 32-elements bitonic sequences (in 4 steps).



5) Then sort the 32 elements bitonic sequences (in 5 steps).

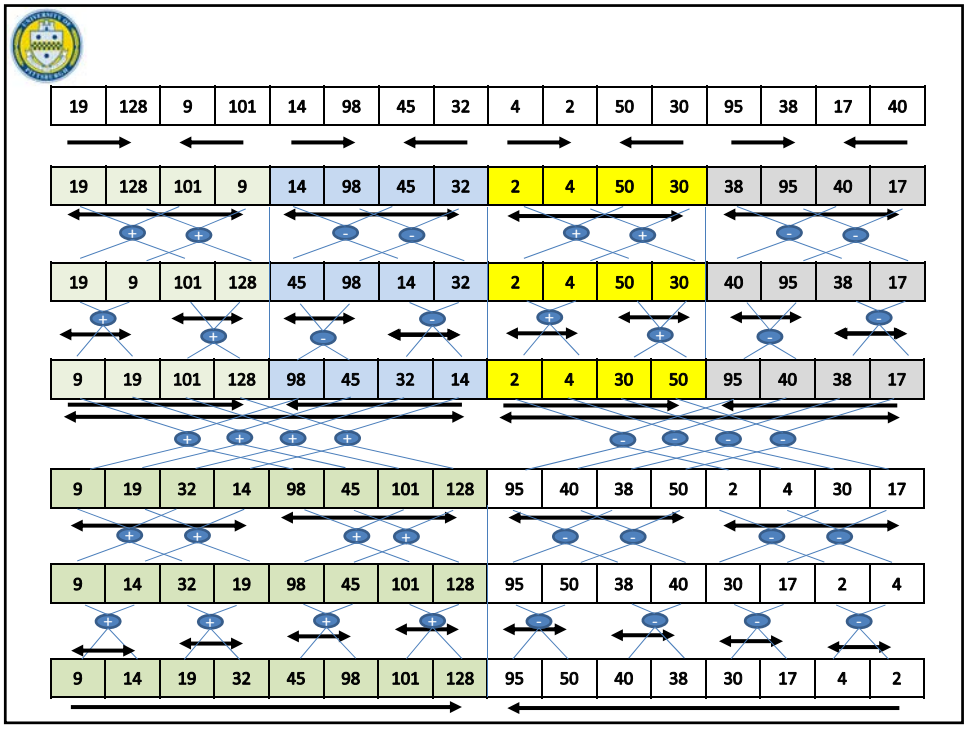


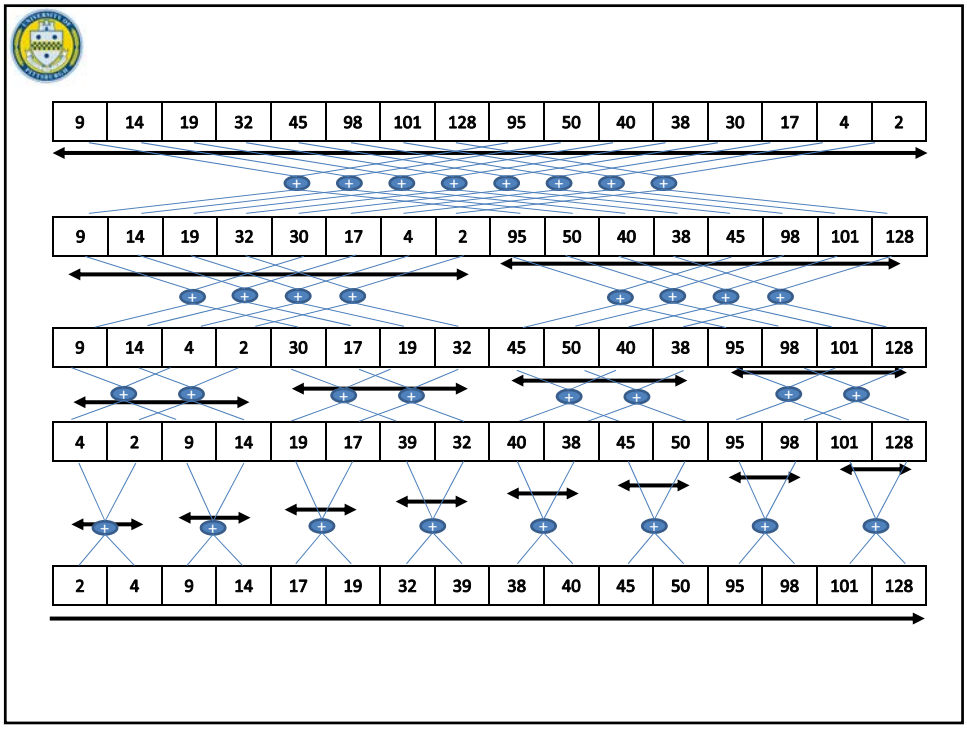
*Computation of execution time:*

*Log n* steps,

Each step, *i*, requires *i* sub-steps. Hence

$$number\_of\_steps = \sum_{i=1}^{\log n} i = \frac{1+\log n}{2}\log n$$


19 128 9 101 14 98 45 32 4 2 50 30 95 38 17 40
19 128 101 9 14 98 45 32 2 4 50 30 38 95 40 17
19 9 101 128 45 98 14 32 2 4 50 30 40 95 38 17
9 19 101 128 98 45 32 14 2 4 30 50 95 40 38 17
9 19 32 14 98 45 101 128 95 40 38 50 2 4 30 17
9 14 32 19 98 45 101 128 95 50 38 40 30 17 2 4
9 14 19 32 45 98 101 128 95 50 40 38 30 17 4 2


9 14 19 32 45 98 101 128 95 50 40 38 30 17 4 2
9 14 19 32 30 17 4 2 95 50 40 38 45 98 101 128
9 14 4 2 30 17 19 32 45 50 40 38 95 98 101 128
4 2 9 14 19 17 39 32 40 38 45 50 95 98 101 128
2 4 9 14 17 19 32 39 38 40 45 50 95 98 101 128