Bubble Sort

Kartik Gopalan
Sequential Bubble Sort
Sequential Bubble Sort

10 5 3 11 0 2 6 8
Sequential Bubble Sort

Compare and Swap

10 5 3 11 0 2 6 8
Sequential Bubble Sort

5 10 3 11 0 2 6 8
Sequential Bubble Sort

5 3 10 11 0 2 6 8
Sequential Bubble Sort

5  3  10  11  0  2  6  8
Sequential Bubble Sort

5  3  10  0  11  2  6  8
Sequential Bubble Sort

5  3  10  0  2  11  6  8
Sequential Bubble Sort

5 3 10 0 2 6 11 8
Sequential Bubble Sort
Sequential Bubble Sort

3  5  10  0  2  6  8  11
Sequential Bubble Sort

3  5  10  0  2  6  8  11
Sequential Bubble Sort

3  5  0  10  2  6  8  11
Sequential Bubble Sort

3 5 0 2 10 6 8 11
Sequential Bubble Sort

3 5 0 2 6 10 8 11
Sequential Bubble Sort

3 5 0 2 6 8 10 11
Sequential Bubble Sort

3 5 0 2 6 8 10 11
Sequential Bubble Sort

3 0 5 2 6 8 10 11
Sequential Bubble Sort

3 0 2 5 6 8 10 11
Sequential Bubble Sort

3 0 2 5 6 8 10 11
Sequential Bubble Sort

3 0 2 5 6 8 10 11
Sequential Bubble Sort

0 3 2 5 6 8 10 11
Sequential Bubble Sort

0 2 3 5 6 8 10 11
Sequential Bubble Sort

0 2 3 5 6 8 10 11
Sequential Bubble Sort

0 2 3 5 6 8 10 11
Sequential Bubble Sort

0 2 3 5 6 8 10 11
Sequential Bubble Sort

0 2 3 5 6 8 10 11
Sequential Bubble Sort

No swaps in the last round.

Sorted!

Stop.
Even-Odd Pass Bubble Sort
Even-Odd Pass Bubble Sort

Compare and swap even pairs (sequentially)
Even-Odd Pass Bubble Sort

Compare and swap odd pairs (sequentially)
Even-Odd Pass Bubble Sort

Even pairs

5 3 10 0 11 2 6 8

0 1 2 3 4 5 6 7
Even-Odd Pass Bubble Sort

Odd pairs

<table>
<thead>
<tr>
<th>3</th>
<th>5</th>
<th>0</th>
<th>10</th>
<th>2</th>
<th>11</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
Even-Odd Pass Bubble Sort

Even pairs

3 0 5 2 10 6 11 8
0 1 2 3 4 5 6 7
Even-Odd Pass Bubble Sort

Odd pairs

0  1  2  3  4  5  6  7  8  9  10  11
Even-Odd Pass Bubble Sort

Even pairs

0  2  3  5  6  8  10  11
Even-Odd Pass Bubble Sort

Odd pairs
Even-Odd Pass Bubble Sort

No swaps in the last two rounds.
Sorted!
Stop.
Parallel
Even-Odd Pass Bubble Sort
Even-Odd Pass Bubble Sort

Concurrent worker processes operate on overlapping segments of the array

Worker 0  Worker 1  Worker 2  Worker 3
Even-Odd Pass Bubble Sort

Concurrent worker processes operate on overlapping segments of the array

Each worker does the following on its range
- Barrier (pass == even)
- Even pass bubbling
- Barrier (pass == odd)
- Odd pass bubbling
Even-Odd Pass Bubble Sort

• Barrier (pass == even)
• Even pass bubbling
Even-Odd Pass Bubble Sort

Worker 0  Worker 1  Worker 2  Worker 3 (does nothing)

- Barrier (pass == odd)
- Odd pass bubbling

And so forth as before…
….but in parallel
Even-Odd Pass Bubble Sort

<table>
<thead>
<tr>
<th>5</th>
<th>10</th>
<th>3</th>
<th>11</th>
<th>0</th>
<th>2</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Worker 0  Worker 1  Worker 2  Worker 3 (does nothing)

Barrier is simply a busy while loop checking for a condition

```
barrier( condition )
{
    while( !condition );
}
```