PARALLEL PACK using Parallel Scan - Prefix Sums - Homework
Due: Wednesday, April 13 - beginning of class

Consider an array integer elements containing some zero elements. The task is to PACK the non-zero elements into a new array, into the low-numbered locations.

For example:

<table>
<thead>
<tr>
<th>Loc:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>3</td>
<td>0</td>
<td>78</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

Packs to:

| 14 | 3 | 78 | -3 | 7 | 0 | 0 | 0 | 0 |

Algorithm:
- Assign 1 to non-zero elements; assign 0 to zero elements
- Compute Scan (Prefix Sum)
- Copy array element into location determined by Scan (Prefix Sum)

Homework:
Consider an array of size n and p processors with n >> p (much larger). You can assume n is a power of 2 and is divisible by p if it helps your thought process.

1. What is the Big Oh time complexity of the Parallel Scan (Prefix Sum) operation in terms of n and p? State the time for each step, explain why then give the total time complexity.

2. What is the Big Oh time complexity of the Parallel Pack (including part 1 above) in terms of n and p? State the time for each step, explain why then give the total time complexity.