This quiz has 3 questions, for a total of 30 points.

1. [10 points] Given the following array, turn it into a max heap, showing the array and tree representation before and after each Max-Heapify operation.

   \[3, 5, 4, 6, 2, 7\]

   **Solution:** Here is the initial heap represented as a binary tree: (1 point)

   ![Binary tree diagram](image)

   After applying Max-Heapify to 4 at index 2: (1 point)

   \[3, 5, 7, 6, 2, 4\]

   ![Binary tree diagram](image)

   After applying Max-Heapify to 5 at index 1: (1 point)

   \[3, 6, 7, 5, 2, 4\]

   ![Binary tree diagram](image)

   After applying Max-Heapify to 3 at index 0: (1 point)

   \[7, 6, 4, 5, 2, 3\]

   ![Binary tree diagram](image)

2. [10 points] Write down an implementation of quicksort in Python. The parameters are an array (Python list) A and the range within A to be sorted, starting with the element at position start and ending just before the position end. You may use the partition function without implementing it. Recall that the partition function has the same parameters as quicksort. The following code shows an example of using quicksort.

   ```python
   A = [2, 8, 7, 1, 3, 5, 6, 4]
   sorted_A = [1, 2, 3, 4, 5, 6, 7, 8]
   quicksort(A, 0, len(A))
   assert A == sorted_A
   
   def quicksort(A, start, end):
   ```
3. **10 points** Given a hashtable with table size 4, insert the keys 0, 1, 2, 3, 4, 5 into the hashtable using the division method, \( h(k) = k \mod 4 \), and chaining to handle collisions. Draw the resulting hashtable. How many collisions occurred during the insertion of the keys? What is the load factor of the resulting hashtable?

**Solution:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Hashed Key</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

There were 2 collisions. 4 collided with 0 and 5 collided with 1. (2 points)

The load factor is \( 6/4 = 1.5 \). (2 points)