1. Consider the divide and conquer algorithm for finding the closest pair of points. Analyze the time complexity of the algorithm. Include and discuss a detailed discussion of how to manage points in the x-dimension and how to manage (and search) points in the y-dimension. (You should do this without consulting the book or your notes.)

2. Exercise 4 on page 315

3. Exercise 6 on page 317

4. Exercise 9 on Page 320

5. Given a rod of length \( n \) inches and an array of prices that contains prices of all pieces of size smaller than \( n \). Determine the maximum value obtainable by cutting up the rod and selling the pieces. For example, if length of the rod is 8 and the values of different pieces are given as following, then the maximum obtainable value is 22 (by cutting in two pieces of lengths 2 and 6)

<table>
<thead>
<tr>
<th>length</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>price</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>17</td>
<td>17</td>
<td>20</td>
</tr>
</tbody>
</table>

6. Consider a row of \( n \) coins of values \( v_1, \ldots, v_n \), where \( n \) is even. We play a game against an opponent by alternating turns (you can both see all coins at all times). In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Determine the maximum possible amount of money we can win if we move first.

   Example 1: [5, 3, 7, 10] : The user collects maximum value as 15(10 + 5)
   - Sometimes the greedy strategy works

   Example 2. [8, 15, 3, 7] : The user collects maximum value as 22(7 + 15)
   - In general the greedy strategy does not work