1. **Java (3):**
   Write a java method that sorts an array using any method. List input, output, preconditions, and post conditions.

2. **Divide and Conquer (10):**
   Write an algorithm which sorts a set of n numbers using at most \( \log(n!) + n \) number of comparisons.
   *(You may use any operation that does not compare a pair of elements as many times as you wish)*

3. **Induction (6):**
   Given a set of \( n \geq 2 \) distinct points on a 2 dimensional plane, show that it is always possible to draw a polygon with \( n \) sides containing all points as vertices, such that no two sides intersect.
   *(Assume that all points lie on a single line)*

4. **Landau Symbols (Big O notation) (12):**
   The notation \( o \) (read: small o) can be interpreted to mean the following:
   If \( f(n) = o(g(n)) \),
   then \( f(n) = O(g(n)) \) but \( g(n) \neq O(f(n)) \)
   
   a) Find a function \( f(n) \) such that:
      \( f(n) = o(n) \)
      \( \log(n) = o(f(n)) \)

   b) Find a function \( f(n) \) such that:
      \( f(n) = o(n^c) \), for any \( c > 0 \)
      \( (\log(n))^k = o(f(n)) \), for any \( k > 0 \)

   c) Find a function \( f(n) \) that uses only the binary operations +, -, \*, /, ^, log, such that \( f(n) = O(n^*n!) \)
      and \( n! / n = O(f(n)) \).
      *(Recall that \( \int_{ln(x)} dx - x \* ln(x) - x \)*)

5. **Quicksort (8):**
   There exists an algorithm which can find the \( k_{th} \) element in a list in \( O(n) \) time, and suppose that it is in place. Using this algorithm, write an in place sorting algorithm that runs in worst case time \( O(n^*\log(n)) \), and prove that it does. Given that this algorithm exists, why is mergesort still used?

6. **ADTs (1):**
   Complete the following table with **optimal** big O running times, given the data structure:

<table>
<thead>
<tr>
<th></th>
<th>Array of size n</th>
<th>Linked list of size n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get ( i_{th} ) entry in list, where ( i ) is any number between 1 and n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concatenate two lists</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>