1. Short Answer (14+6+5=25 marks)

Provide your short answers in the provided boxes. There is no need to justify your answers for True/False questions.

1. True or False: An abstract class can implement an interface.  
   
2. A private method of class A can be called in classes inherited from A.  
   
3. True or False: $2019n \in O(n)$  
   
4. True or False: $n^2 \in O(n \log n)$  
   
5. True or False: merge-sort runs in $O(n^2)$ in the worst case  
   
6. Consider the following pseudocode:
   ```java
   void foo (int n)
   {
     int i = 1;
     int q = 1;
     while (i < n)
     {
       for (int j = 1; j < n; j++)
         q = q*j;
       i = i+2;
     }
   }
   ```
   What is the worst-case running time of $foo(n)$? Express your answer using $O$-notation in terms of $n$, and be as precise as possible. No justification is needed.  
   
7. Consider the following figure which plots three functions $f$ and $g$.
   True or False: $f(n) \in O(g(n))$.  

   ![Diagram](attachment:image.png)
8. Consider the following recursive algorithm.

```c
void foo (int n, int k)
{
    if (k < 0)
        return -1; // wrong input
    if (k == 1)
        return n;
    return n * foo (n, k-1);
}
```

a) In one or two sentences indicate what the algorithm does.

b) Indicate whether the code uses a tail recursion or not. In either case, shortly explain why.

9. Assume you want to sort a set of non-negative real numbers whose fractional part is 0 or 0.5. An example is
   \{0.0, 2.5, 3.0, 0.5, 5.5, 8.0, 9.0\}. Indicate whether it is possible to sort them in $O(n)$. Justify your answer in a few sentences.
2. Recursion (7 marks)

For a given integer \( n \) such that \( n \geq 2 \), let’s define \( EF(n) \) as the product of all positive, even integers smaller than or equal to \( n \). For example, \( EF(2) = 2 \), \( EF(3) = 2 \), \( EF(4) = 2 \times 4 \) and \( EF(9) = 2 \times 4 \times 6 \times 8 \).

Write a recursive code to compute and return \( EF(n) \). You need to complete the following code:

```java
public int EF(int n) {
    if (n < 2)
        return -1;
    // complete the code in the space below
```
3. More Recursion (7 marks)

The goal of this question is to devise a code that recursively reverses a string. For example, given the string “wolf rats”, it should be converted to “star flow”.

You should complete the following code:

```java
public void reverse(String str)
{
    reverse_rec(str, 0, str.length());
}

private void reverse_rec(String str, int lo, int hi)
{
    // implement this. You should recognise what lo and hi are
```
4. Sorting (8+4=12 marks)

Consider the quick sort algorithm for sorting $n$ integers. Assume you are given a pivot selection algorithm that guarantees a pivot $p$ such that at least $n/3$ numbers in the array are larger than $x$ and at least $n/3$ items are smaller than $x$.

a) Drive a recursive function for the best-case time complexity of the quick-sort with this pivot-selection algorithm.

b) Drive a recursive function for the worst-case time complexity of the quick-sort with this pivot-selection algorithm.

c) (bonus) What is the time complexity of the algorithm in the worst case? To get the complete mark, you need to justify your answer. It is not sufficient to provide a correct time without a correct justification.