1. Consider this Partial Sort algorithm that repeatedly moves the first element in array A to its correct place if the array were sorted, and stops when it finds a first element that is already in its correct place.

```
 PS(A)
 1. k := A.length - 1
 2. while (k > 0)
 3.     i := 0
 4.     j := k
 5.     p := A[0]
 6.     while (j > i)
 7.         if (A[j] < p)
 8.             A[i] := A[j]
 9.             i := i + 1
10.             A[j] := A[i]
            else
11.             j := j - 1
12.     A[i] := p
13.     k := i - 1
```

   (a) Compute the exact number of steps executed for each line of the algorithm.

   (b) Compute $t_{PS}([4,2,1,3,5])$ exactly.

   (c) Compute $T_{PS}(3)$ exactly.

   (d) Prove that $T_{PS}(n) \in \Theta(n^2)$, where $n =$A.length.

2. Consider a normalized floating point system with base $\beta = 2$, mantissa size $t = 3$, minimum and maximum exponents $e_{min} = -2$ and $e_{max} = +3$. Plot and label all the nonnegative numbers of this system on a real number line that is drawn to scale.

3. Consider the function f : [-π,π] $\rightarrow \mathbb{R}$ defined by:

   f(x) = 1 - cos x

   (a) Explain why it is unstable to compute f(x) using this Java statement:

```
return 1 - Math.cos(x);
```

   (b) Find a more stable way to compute f(x). Explain your answer.
   Hint: Recall the trigonometric identity: $sin^2\, x + cos^2\, x = 1$