Exercises

1. An algorithm **INSERT** for inserting a value into a sorted array could be defined by the following specifications:

   **Inputs** Array \( A = [a_1, \ldots, a_n] \) of numbers in sorted order (low to high); a number \( v \)

   **Output** Array \( B \) of \((n + 1)\) numbers in sorted order, consisting of the numbers \( a_1, \ldots, a_n \) and \( v \)

   For example, \( \text{Insert}([1,2,4],3) \) would return \([1,2,3,4]\).

   (a) Imagine that you’re given an **INSERT** subroutine to use in your algorithms. Give an algorithm that uses that subroutine to sort an input array. **Be sure to describe your algorithm in English!** You can also give pseudocode as part of a clear, detailed description of the algorithm, but a pseudocode-only presentation will not receive full credit.

   (b) Although we can’t know the complexity of the **INSERT** subroutine, what do you believe an upper bound on its time complexity would be? Give a thorough explanation of your answer!

2. Exercise 34-1 in CLRS (pages 1101–1102) defines the independent-set problem. As presented, it is an optimization problem. Formulate a related **independent-set decision problem** by giving the input / output specifications of the decision variant.

3. The **bin-packing problem** is an optimization problem specified as follows:

   **Input** Set \( S = \{i_1, \ldots, i_n\} \) of \( n \) items, where item \( i_z \) has associated rational-number size \( s_z \) (\( 0 < s_z \leq 1 \))

   **Output** Integer \( m \), which is the smallest integer such that all items in \( S \) can fit into \( m \) bins of size 1.
For example, if $S$ has three items all with size $\frac{1}{2}$, they could fit in 2 bins of size 1—two items in one bin, one item in the third. If $S$ has five items of sizes $\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}$, they could fit in three bins of size 1, but they could not fit in two bins of size 1.

Formulate a related bin-packing decision problem by giving the input / output specifications of the decision variant.