```
MERGE-SORT(A[1..n], n)
  if n > 1 then
    m ← ⌊n/2⌋
    A' ← A[1..m]
    A'' ← A[m+1..n]
    MERGESORT(A', m)
    MERGESORT(A'', n − m)
    A ← MERGE(A',m,A'',n-m)
```

Example 1. Example 1.

Consider the algorithm:

A[1] ← k

return 1

or

k ← A[1]

return 1

1 Correctness of Algorithms

An algorithm is correct if it satisfies its specifications

Specifications are often written using preconditions (certain facts must be true before an execution of the algorithm begins, it can describe what inputs are allowed) and postconditions (certain facts must be true when an execution of the algorithm ends, often it describes the correct ouput or possible correct outputs for a given input), and termination - the algorithm halts when the preconditions are true

Example 2. Specifications for Binary Search Algorithm
Precondition: $A[1..nn]$ is sorted in non decreasing order

$$\forall i \in \mathbb{Z}^+ . \forall j \in \mathbb{Z}^+ . [(i < j \leq n) \text{ IMPLIES } (A[i] \leq A[j])]$$

[a]

Postcondition: same as for searching an array

[a]elements of $A[1..n]$ and k must be from a totally ordered domain.

2 Sorting an array

Precondition: the elements in A[1..n] are from a totally ordered domain

Postcondition:

- The multiset of elements in A[1..n] is not changed
- The elements in A[1..n] afterwrads are a permutation of the elements in A[1..n] before the algorithm was in A[1..n] before the algorithm has executed
- The elements of A are in nondecreasing order.

3 Merging two arrays

Preconditions: A[1..m] and B[1..n] are sorted in nondecreasing order. The elements in A[1..m] and B[1..n] are from a totally ordered domain.

Postcondition: Outputs an array C[1..$m + n$] such that the multiset of elements in C[1..$m + n$] is the union of the multisets of elements in A[1..m] and B[1..n]. A and B are not changed. C is sorted in nondecreasing order.