

## Program 1

CS 240  
Fall 1992  
Craig A. Rich

The following is an Ada generic package specification which provides the six sorting algorithms we have studied in class.

sort\_.ada

```
generic
  type ELEMENT is private;
  type VECTOR  is array (POSITIVE range <>) of ELEMENT;
  with function "<" (LEFT, RIGHT: ELEMENT) return BOOLEAN;

package SORT is

  procedure BUBBLE_SORT      (V: in out VECTOR);
  procedure INSERTION_SORT  (V: in out VECTOR);
  procedure SELECTION_SORT  (V: in out VECTOR);
  procedure QUICK_SORT      (V: in out VECTOR);
  procedure HEAP_SORT       (V: in out VECTOR);
  procedure MERGE_SORT      (V: in out VECTOR);

end SORT;
```

1. Create the package specification `sort_.ada` and the package body `sort.ada` which contains the implementation of each sorting algorithm.
2. Create the procedure `program_1.ada` which inputs at most 10000 natural numbers from the standard input file into a vector and sorts the vector into ascending order using each of the six sorting algorithms. Measure the running time of each algorithm and output the following information to the standard output file:

```
Length of vector is 5857
Bubble Sort      43.9400 seconds
Insertion Sort   30.6000 seconds
Selection Sort   32.3300 seconds
Quick Sort       0.2400 seconds
Heap Sort        1.3900 seconds
Merge Sort       0.6400 seconds
```

3. The predefined package `CALENDAR` provides types `TIME` and `DURATION`, and function `CLOCK` which are suitable for measuring the running time of the sorting algorithms. The following procedure gives an example of their use:

```
with TEXT_IO, CALENDAR;
use TEXT_IO, CALENDAR;

procedure INTERVAL is

    package DURATION_IO is new FIXED_IO(DURATION);
    use DURATION_IO;

    START: TIME;

begin
    START := CLOCK;
    :
    PUT (CLOCK-START);
end INTERVAL;
```

4. Hand in one contiguous printout containing compiler listings of the files `sort_ada`, `sort.ada`, and `program_1.ada`, and a listing of the output produced by `PROGRAM_1` from input file `[cs.carich]program_1.dat`—in that order.