Algorithms 2023/24

Degree in Computer Science Engineering

Practical 2

Submission deadline: Saturday, 21st October, 23:59

Insertion sort and Shell sort: The problem consists in sorting an array of n integers in ascending order. *Insertion sort* and *Shell sort* will be used as sorting algorithms:

```
procedure Insertion sort (var v[1..n])
  for i := 2 to n do
    x := v[i] ;
    j := i-1 ;
    while j > 0 and v[j] > x do
      v[j+1] := v[j] ;
      j := j-1
    end while ;
    v[j+1] := x
  end for
end procedure

procedure Shell Sort (v[1..n])
  increment := n;
  repeat
    increment := increment div 2;
    for i := increment+1 to n do
      tmp := v[i];
      j := i;
      keepgoing := true;
      while j-increment > 0 and keepgoing do
        if tmp < v[j-increment] then
          v[j] := v[j-increment];
          j := j-increment
        else  keepgoing := false
        end if
      end while;
      v[j] := tmp
    end for
  until increment = 1
end procedure
```

1. Implement the insertion sort and Shell sort algorithms.

   ```
   void ins_sort (int v [], int n);
   void shell_sort (int v [], int n);
   ```

2. Validate the correct functioning of the implementation.

   ```
   > ./test
   Random initialization
   3, -3, 0, 17, -5, 2, 11, 13, 6, 1, 7, 14, 1, -2, 5, -14, -2
   sorted? 0
   Insertion sort
   -14, -5, -3, -2, -2, 0, 1, 1, 2, 3, 5, 6, 7, 11, 13, 14, 17
   sorted? 1
   ```

```
Insertion sort with descending initialization
           n           t(n)     t(n)/n^1.8       t(n)/n^2     t(n)/n^2.2
(*)      500         247.03       0.003425       0.000988       0.000285
        1000         953.00       0.003794       0.000953       0.000239
        2000        3818.00       0.004365       0.000955       0.000209
        4000       15471.00       0.005079       0.000967       0.000184
        8000       69474.00       0.006550       0.001086       0.000180
       16000      257089.00       0.006961       0.001004       0.000145
       32000     1023540.00       0.007959       0.001000       0.000126
```

Figure 1: Part of the possible output to screen of the main program's execution

```
Descending initialization
10, 9, 8, 7, 6, 5, 4, 3, 2, 1
sorted? 0
Insertion sort
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
sorted? 1
```

3. Determine the execution times for different values of n and for three different initial scenarios: (a) the array is already sorted in ascending order, (b) the array is already sorted in descending order, and (c) the array is initially disordered.

4. Empirically calculate the complexity of each of the algorithms for each of the initial scenarios of the array (i.e., 6 tables) (figure 1).

5. Submit the C code files and the `.txt` file with the report using the task *Practical 2 Submission* at the Algorithms page in `https://campusvirtual.udc.gal`. We remind you that the deadline to complete the task is on Saturday, 21st of October, at 23:59, and once submitted, the files cannot be changed. All the students in a team must submit the work.