

**Practical 4: Heaps**

Submission deadline: Wednesday, 7 December

1. Implement the following operations on **max heaps**.

```
#define SZ 512000
typedef struct {
    int vector[SZ];
    int last;
} heap;

void create_heap(int [], int, heap *);
int remove_max(heap *);
```

Validate that the operations `create_heap` and `remove_max` work correctly.

Note: since arrays in C are indexed starting with zero, the relative positions of the children and parent of a node in the array have to be recalculated with respect to the ones seen in the theoretical lessons.

2. Show empirically that creating a heap from any given array with  $n$  elements (operation `create_heap`) runs in time  $O(n)$ . To this end, we recommend using ascending sequences of numbers.
3. Implement *heap sort*: `void heap_sort(int [], int)`

```
procedure Heap sort (var A[1..n])
    create_heap (A, H);
    for i := n to 1 by -1 do
        A[i] := remove_max(H);
    end for
end procedure
```

Validate that the sorting algorithm works correctly.

4. Empirically calculate the complexity of the sorting algorithm 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 unsorted.
5. Submit the files with the C source code and the .txt file with the report by means of the task *Practical 4 Submission* in the Algorithms website at <https://campusvirtual.udc.gal>. Remember that the deadline to complete the task expires on Wednesday, 7th December, at 23:59. Once uploaded, files cannot be changed. **The work has to be submitted by all the members of each team.**