Advanced algorithmic : Addendum to Practical session #1

Data Structures and Complexity

2 Supervised Practical Session + 1 Unsupervised Practical Session + 1 Quiz

The HeapSort algorithm HeapSort(T) takes an array T and does a BuildHeap(T) in order to obtain a **maximum** binary heap, then it does n successive RemoveBis on T, where RemoveBis is a particular operation that exchanges the root with the last element (instead of only replacing the root by this last element) then PercolateDown the root, then decrease T.Size. After the first RemoveBis, the greatest element is at the last place of the array ($T[n]$) but the size of the heap T.Size is decreased (T.Size$= n-1$), the capacity of the heap remaining the same (T.Length$= n$).

Supplementary Work to do before next supervised session

1. implement HeapSort
2. implement the solution of sorting all the data with HeapSort then extract the k greatests.
3. Compare experimentally the efficiency Quicksort to HeapSort for a value of n great enough (10^7 for instance).
4. Modify your algorithms to count the number of swaps that are done. Compare the number of swaps done by BuildHeap to the number of swaps done by adding successively the n elements by Add.