Advanced Programming Techniques

Aurel Vlaicu University Arad

Practical Session #4

In the following, the objective is to code, compare and analyze the following sorting algorithms:

1. InsertionSort
2. QuickSort
3. MergeSort
4. HeapSort
5. others

Exercise 1. (Experimental analysis) (a) Implement all these sorting algorithms.

(b) Let N be the size of the array to be sorted. Compute empirically the time the algorithm takes compared to N:

N	InsertionSort	QuickSort	MergeSort	HeapSort
10				
100				
1000				
10000				
100000				
1000000				

For each N run the sorting algorithm a fixed number k of times and compute the average running time.

(c) Analyze the results: conclude which algorithms are faster experimentally for:

- small values of N
- large values of N

(d) Perform the same analysis for the worst case scenario.

Exercise 2. Comb Sort.

Consider the following sorting algorithm which compares elements far away.

- gap = $n - 1$
- for i=1: n-gap+1
- if $A[i] > A[i + gap]$ then swap them
- Decrease the gap by a factor (gap = floor(gap/1.3))
- if $gap < 1$ then gap=1.
- repeat until no swaps are made.

(a) Implement comb sort. Evaluate its complexity (search online references if needed)

(b) Compare it with previously described algorithms.