**Heap sort**

**Heap sort** is an efficient sorting algorithm implemented with the [heap](http://www.algorithmist.com/index.php?title=Heap&action=edit&redlink=1) data structure.

## Algorithm

1. Build a heap with the sorting array, using recursive insertion.
2. Iterate to extract n times the maximum or minimum element in heap and heapify the heap.
3. The extracted elements form a sorted subsequence.

## Pseudocode

Heapsort(A as array)

BuildHeap(A)

for i = n to 1

swap(A[1], A[i])

n = n - 1

Heapify(A, 1)

BuildHeap(A as array)

n = elements\_in(A)

for i = floor(n/2) to 1

Heapify(A,i)

Heapify(A as array, i as int)

left = 2i

right = 2i+1

if (left <= n) and (A[left] > A[i])

max = left

else

max = i

if (right<=n) and (A[right] > A[max])

max = right

if (max != i)

swap(A[i], A[max])

Heapify(A, max)

## Implementations

[C](http://www.algorithmist.com/index.php/Heap_sort.c) in-place non-recursive

Example  
  


