COUNTING SORT(A,B,k) algorithm

1 for i = 1 to k

2 do c[i] = 0

3 for j = 1 to length[A]

4 do C[A[j]] = C[A[j]] + 1

5 >C[i] now contains the number of elements equal to i

6 for i = 2 to k

7 do C[i] = C[i] + C[i-1]

8 >C[i] now contains the number of elements less than or equal to i

9 for j = length[A] downto 1

10 do B[C[A[j]]] = A[j]

11 C[A[j]] = C[A[J]] - 1

link   
<http://www.cse.iitk.ac.in/users/dsrkg/cs210/applets/sortingII/countingSort/algCount.html>

**Heapsort**

Heapsort(A) {

BuildHeap(A)

for i <- length(A) downto 2 {

exchange A[1] <-> A[i]

heapsize <- heapsize -1

Heapify(A, 1)

}

BuildHeap(A) {

heapsize <- length(A)

for i <- floor( length/2 ) downto 1

Heapify(A, i)

}

Heapify(A, i) {

le <- left(i)

ri <- right(i)

if (le<=heapsize) and (A[le]>A[i])

largest <- le

else

largest <- i

if (ri<=heapsize) and (A[ri]>A[largest])

largest <- ri

if (largest != i) {

exchange A[i] <-> A[largest]

Heapify(A, largest)

}

}