Algorithm deleteKeyHeapBasedPriorityQueue(arrHeap)

Input: Array which build heap

Output: Array which deleted a key from a heap-based priority queue.

n <- arrHeap.length

arrHeap[0] <- arrHeap[n-1]

n <- n-1

newArrHeap <- new int[n];

for i <- 0 to arrHeap.length -1 do

newArrHeap <- arrHeap[i]

if n > 0 then

downHeap(0, newArrHeap)

return newArrHeap

Algorithm downHeap(index, arrHeap)

Input: index of parent, array which build heap

Output: A down heap

leftChild <- 0

rightChild <- 0

min <- 0

n <- arrHeap.length -1

if rightChild >= n do

if leftChild >= n then

return

else

min = leftChild

else

if arrHeap[leftChild] <= arrHeap[rightChild] then

min = leftChild

else

min = rightChild

if arrHeap[index] > arrHeap[min] then

swap(arrHeap[index], arrHeap[min])

downHeap(min, arrHeap)