Heapsort Sample Code

 def heapsort( aList ):

  # convert aList to heap

  length = len( aList ) - 1

  leastParent = length / 2

  for i in range ( leastParent, -1, -1 ):

    moveDown( aList, i, length )

  # flatten heap into sorted array

  for i in range ( length, 0, -1 ):

    if aList[0] > aList[i]:

      swap( aList, 0, i )

      moveDown( aList, 0, i - 1 )

def moveDown( aList, first, last ):

  largest = 2 \* first + 1

  while largest <= last:

    # right child exists and is larger than left child

    if ( largest < last ) and ( aList[largest] < aList[largest + 1] ):

      largest += 1

    # right child is larger than parent

    if aList[largest] > aList[first]:

      swap( aList, largest, first )

      # move down to largest child

      first = largest;

      largest = 2 \* first + 1

    else:

      return # force exit

def swap( A, x, y ):

  tmp = A[x]

  A[x] = A[y]

  A[y] = tmp