HEAP SORT

```
Algorithm ReHepifyDown(H, start, finish)
{
      LChild := 2 * start;
      RChild := 2 * start + 1;
      if(LChild <= finish) then
      {
            max := H[Lchild];
            index := LChild;
            if(RChild <= finish) then
            {
                  max := H[RChild];
                  index := RChild;
            }
            if(H[start] < H[index])
            {
                  swap(H[start], H[index]);
                  ReHepifyDown(H, index, finish);
            }
      }
}

Algorithm Hepify(H, n)
{
      for i := n/2 to 1 by -1 do
            ReHepifyDown(H, i, n);
}

Algorithm HeapSort(H, n)
{
      Hepify(H, n);
      for i := n to 1 by -1 do
      {
            swap(H[1], H[i]);
            ReHepifyDown(H, 1, i-1);
      }
}
```