

Assignment 2 Problem 1

A *stable* sorting algorithm is one in which the relative order of all identical elements (or keys) is the same in the output as it was in the input. Prove, using a well-chosen example, that heapsort is not a stable sorting algorithm.

solution:

Let $a = 0$, $b = 1$, $c = 1$, $d = 2$ $e = 3$ $f = 4$ and $g = 5$

Let the original array A to be $A[7] = \{g, f, e, d, c, b, a\}$

Suppose the heapsort is a stable sorting, then the output should be $A[7] = \{a, c, b, d, e, f, g\}$

Assuming the HeapSort we used have the following pseudocode

HeapSort(A, n) (1)

$n = A.size$ (2)

for $i = parent(last(n))$ down to 0 do (3)

fix-down(A, n, i) (4)

while $n > 1$ (5)

 swap items at $A[root()]$ and $A[last(n)]$ (6)

 decrease n (7)

fix-down($A, n, root()$) (8)

(9)

fix-down(A, n, k) (10)

while k is not a leaf do (11)

$j = left-child(k)$ (12)

 if ($j = last(n)$ and $A[j + 1] > A[j]$) (13)

$j = j + 1$ (14)

 if ($A[k] \geq A[j]$) break (15)

 swap($A[j], A[k]$) (16)

$k = j$ (17)

(18)

After doing the heapify part of the code (ie. line 2 to line 4 in the previous pseudocode), the entries in array A is now $\{g, f, e, d, c, b, a\}$ which does not change since it is already a maximum heap.

Then, we repeatedly find maximum(ie. line 5 to line 8) after the 1st loop, we get $A[7] = \{f, d, e, a, c, b, g\}$

after the 2nd loop, we get $A[7] = \{e, d, b, a, c, f, g\}$

after the 3rd loop, we get $A[7] = \{d, c, b, a, e, f, g\}$

after the 4th loop, we get $A[7] = \{c, a, b, d, e, f, g\}$

after the 5th loop, we get $A[7] = \{b, a, c, d, e, f, g\}$

after the 6th loop, we get $A[7] = \{a, b, c, d, e, f, g\}$ which is also the final output

Notice that this is not identical to our expected output, therefore, we can conclude that the heapsort is not a stable sorting algorithm.

QED.