

Heap():

Abstract Object:

{Pre: $\{e_1, e_2, e_3, e_4, \dots, e_n\}$ heapsize, size}



Invariant: $\{e_i > e_{i+1} > e_{i+2} \vee e_i < e_{i+1} < e_{i+2} \text{, heapsize} > 0, \text{size} = \text{heapsize}\}$

Primitive Operations:

CreateHeap(): \rightarrow Heap Creates an empty Heap [constructor]

Heapify(item): \rightarrow void Orders the Heap from a given node [Modify]

Build-Heap(): \rightarrow void Orders the Heap starting from the middle index [Modify]

HeapSort(): \rightarrow Array Orders the array [Modify]

CreateHeap():

"Creates an empty Heap"

{Pre: }

{Pos: $\{e_1, e_2, e_3, e_4, \dots, e_n\}$ heapsize, size}

Heapify():

"Orders the Heap and the branch from a given node"

{Pre: $\{e_1, e_2, e_3, e_4, \dots, e_n\}$ }

{Pos: The branch from the given index $_j$ now fulfills the following order property
 $e_i > e_{i+1} > e_{i+2} \vee e_i < e_{i+1} < e_{i+2}$ }

Build-Heap():

"Orders the Heap starting from the middle index"

{Pre: $\{e_1, e_2, e_3, e_4, \dots, e_n\}$ }

{Pos: $\{e_i > e_{i+1} > e_{i+2} \vee e_i < e_{i+1} < e_{i+2}\}$ }

HeapSort():

"Orders the array"

{Pre: $\{e_1, e_2, e_3, e_4, \dots, e_n\}$ }

{Pos: $\{e_1, e_2, e_3, e_4, \dots, e_n \vee e_n, \dots, e_4, e_3, e_2, e_1\}$ }