NAME →SHIVAM KUMAR

ID→ 201551087

REPORT→ ALGO LAB-6

HEAP

COMMENTS→

Max-Heapify()

// Input: A: an array where the left and right children of i root heaps (but i may not), i: an array index

// Output: A modified so that i roots a heap

// Running Time: O(log n) where n = heap-size[A] – i.

Build-Max-Heap()

// Input: A: an (unsorted) array

// Output: A modified to represent a heap.

// Running Time: O(n) where n = length[A]

Heap-Sort()

// Input: A: an (unsorted) array

// Output: A modified to be sorted from smallest to largest

// Running Time: O(n log n) where n = length[A]

__Heap-Increase-Key()__

// Input: A: an array representing a heap, i: an array index, key: a new key greater than A[i]

// Output: A still representing a heap where the key of A[i] was increased to key

// Running Time: O(log n) where n =heap-size[A]

__Heap-Extract-Max()__

// Input: A: an array representing a heap

// Output: The maximum element of A and A as a heap with this element removed

// Running Time: O(log n) where n =heap-size[A]

__Max-Heap-Insert()__

// Input: A: an array representing a heap, key: a key to insert

// Output: A modified to include key

// Running Time: O(log n) where n =heap-size[A]