Eclipse, Kliezl P. Exercise 6.4 – 5

Q:

a. Design an efficient algorithm for finding and deleting an element of the smallest value in a heap and determine its time efficiency.
b. Design an efficient algorithm for finding and deleting an element of a given value *v* in a heap *H* and determine its time efficiency.

A:

a. Assume that the heap is implemented as an array *H* with parent node keys in the first $\lfloor n/2 \rfloor$ positions of the array, while the leaf node keys in the last $\lceil n/2 \rceil$ positions.

   Since the smallest value of a heap is found in one of its leaf nodes, the smallest element can be searched among the last half of the array *H*. To delete the element:
   1. Swap it with the last element $H[n]$
   2. Decrease one from the size of the heap
   3. Move the swapped element on a higher level from its new position until it is smaller than the element of its new parent node

   Searching for the smallest element in the last half of the array has a time efficiency of O(n), while deleting the element has a time efficiency of of O(log n).

b. Assume the given value *v* is found in position *i* of array *H*

   Finding the given value *v* can be done by linear search. Deleting a certain element of a heap is done similarly to deleting the root:
   1. Swap $H[i]$ with last element $H[n]$
   2. Decrease one from the size of the heap
   3. Move the swapped element either on a higher level or lower level until it is smaller than the element of its new parent node

   Searching for the given element has a time efficiency of O(n), while deleting the element has a time efficiency of of O(log n).