Lab 05 Task

Take input from “in.txt” file.

The first line of the input file gives the number of elements in the unsorted array. The second line gives that number of integer values.

Sample Input:

20

802 327 219 415 648 783 250 891 232 822 604 123 138 505 883 224 86 681 51 310

Sample Output:

The unsorted array is:

802 327 219 415 648 783 250 891 232 822 604 123 138 505 883 224 86 681 51 310

The heap sorted array is:

51 86 123 138 219 224 232 250 310 327 415 505 604 648 681 783 802 822 883 891

The array before increasing key:

51 86 532 138 219 224 232 250 310 327 415 505 604 648 681 783 802 822 883 891

The array after increasing key:

51 86 138 219 224 232 250 310 327 415 505 532 604 648 681 783 802 822 883 891

The array before inserting the key:

51 86 138 219 224 232 250 310 327 415 505 532 604 648 681 783 802 822 883 891 123

The array after inserting the key:

51 86 123 138 219 224 232 250 310 327 415 505 532 604 648 681 783 802 822 883 891

Extracting the maximum value: 891

Pseudocodes

max-heapify(*A*, *i*)
1 *l* *left*(*i*)
2 *r* *right*(*i*)
3 **if** *l* _ *heap-size*[*A*] and *A*[*l*] > *A*[*i*]
4 **then** *largest* *l*
5 **else** *largest* *i*
6 **if** *r* _ *heap-size*[*A*] and *A*[*r*] > *A*[*largest*]
7 **then** *largest* *r*
8 **if** *largest* 6= *i*
9 **then** exchange *A*[*i*] $ *A*[*largest*]
10 max-heapify(*A*, *largest*)

heap-increase-key(*A*, *i*, *key*)
1 **if** *key* < *A*[*i*]
2 **then error** "new key is smaller than current key"
3 *A*[*i*] *key*
4 **while** *i* > 1 and *A*[parent(*i*)] < *A*[*i*]
5 **do** exchange *A*[*i*] $ *A*[*parent*(*i*)]
6 *i* parent(*i*)

max-heap-insert(*A*, *key*)
1 *heap-size*[*A*] *heap-size*[*A*] + 1
2 *A*[*heap-size*[*A*]] *key*
3 *i* *heap-size*[*A*]
4 **while** *i* > 1 and *A*[parent(*i*)] < *A*[*i*]
5 **do** exchange *A*[*i*] $ *A*[*parent*(*i*)]
6 *i* *A*[parent(*i*)]

build-max-heap(*A*)
1 *heap-size*[*A*] 1
2 **for** *i* 2 **to** *length*[*A*]
3 **do** max-heap-insert(*A*,*A*[*i*])

heapsort(*A*)
1 build-max-heap(*A*)
2 **for** *i* *length*[*A*] **downto** 2
3 **do** exchange between *A*[1] and *A*[*i*]
4 *heap-size*[*A*] *heap-size*[*A*] − 1
5 max-heapify(*A*, 1)