Assignment-5

(1) Insertion Sort Algorithm.

InsertionSort(array)

mark first element as Sorted

for each unsorted element X

'extract' the element X

for J <- last Sorted Index down to 0

if (current element J > X

move Sorted element to the right by 1

break loop and insert X there.

end insertionSort.

(2) Suppose the elements are pushed on the stack. in the order {4, 2, 1, 18}

{4, 2, 1, 18}

(2) Step 1 : Push 4, Current max: 4

Step 2: Push 2, Current max: 4

Step 3: push 14, Current max: 14

Step 4: push 1, Current max: 14

Step 5: push 18, Current max: 18

Step 6: pop 18, Current max: 14

(3) Step 1 : Start with array m of size k

Step 2 : m[0] = first element

Step 3 : merge ~~sort~~ each additional element in O(n) running time after each push

Step 4 : find Smallest element in O(1) time, m[0]

If the Smallest element k is popped (m[0] == k), pop m[0] from m, Then the next Smallest element will be m[0] = m[1] and so on.