

Input A =	8	17	12	15	92	16	11	52	41
-----------	---	----	----	----	----	----	----	----	----

Heapsort(A)

```

Build-Max-Heap(A)
for i = length[A] downto 2
do exchange A[1] and A[i]
heap size[A] = heap -size[A] - 1
Max-Heapify(A,1)
  
```

Build-Max-Heap(A)

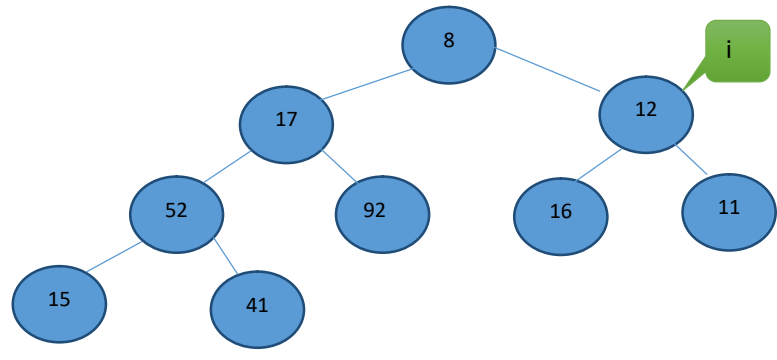
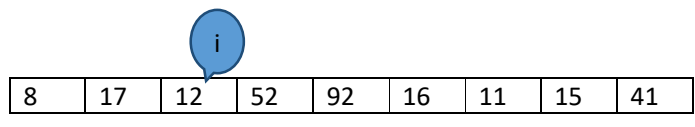
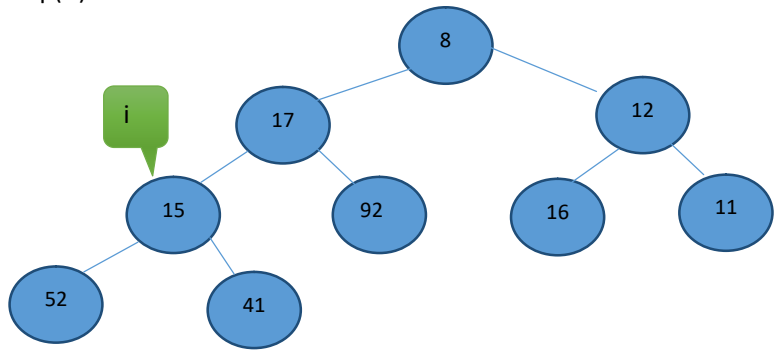
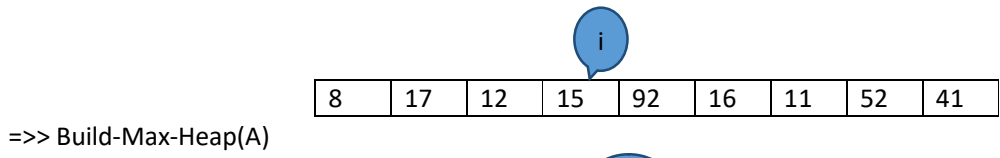
```

heap-size[A] = length[A]
for i = [length[A]/2] downto 1
do Max-Heapify(A,i)
  
```

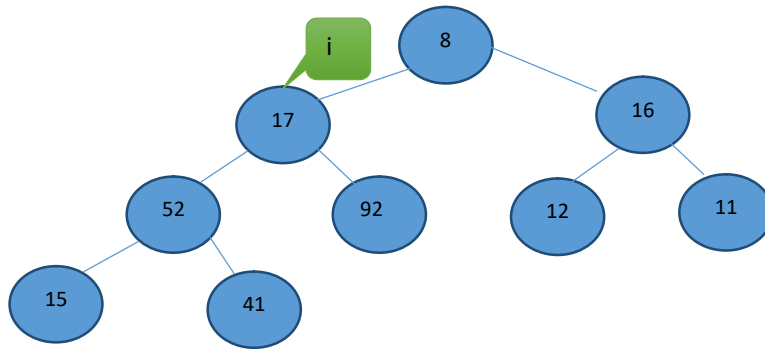
Max-Heapify(A,i)

```

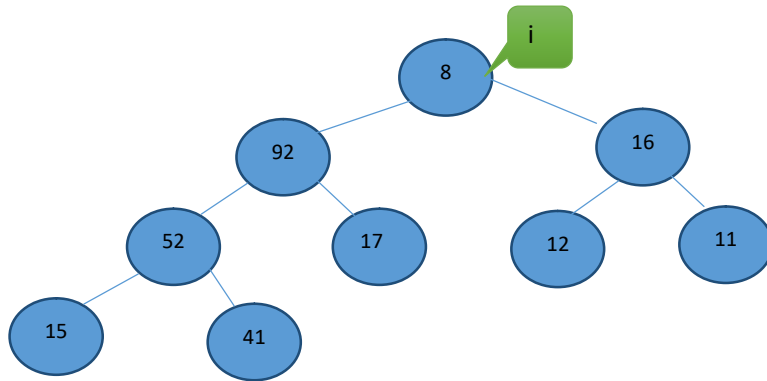
l = left(i)
r = right(i)
if l ≤ heap-size[A] and A[l] > A[i]
  then largest = l
else largest = i
if r ≤ heap-size[A] and A[r] > A[largest]
  then largest = r
If largest != i
  then exchanged A[i] and A[largest]
Max-Heapify(A,largest)
  
```



8	17	16	52	92	12	11	15	41
---	----	----	----	----	----	----	----	----



8	92	16	52	17	12	11	15	41
---	----	----	----	----	----	----	----	----



92	8	16	52	17	12	11	15	41
92	52	16	8	17	12	11	15	41
92	52	16	41	17	12	11	15	8

⇒ Heapsort (code line2-5)

92	52	16	41	17	12	11	15	8
52	41	16	15	17	12	11	8	92

52	41	16	15	17	12	11	8	92
41	17	16	15	8	12	11	52	92

i

41	17	16	15	8	12	11	52	92
17	15	16	11	8	12	41	52	92

i

17	15	16	11	8	12	41	52	92
16	15	12	11	8	17	41	52	92

i

16	15	12	11	8	17	41	52	92
15	11	12	8	16	17	41	52	92

i

15	11	12	8	16	17	41	52	92
12	11	8	15	16	17	41	52	92

i

12	11	8	15	16	17	41	52	92
11	8	12	15	16	17	41	52	92

i

11	8	12	15	16	17	41	52	92
8	11	12	15	16	17	41	52	92