R-2.8

	Running Time
22 15 26 44 10 3 9 13 29 25	10 {insertion of 10 items in unsorted sequence}
3 15 26 44 10 22 9 13 29 25	9 {comparisons are needed}
3 9 26 44 10 22 15 13 29 25	8
3 9 10 44 26 22 15 13 29 25	7
3 9 10 13 26 22 15 44 29 25	6
3 9 10 13 15 22 26 44 29 25	5
3 9 10 13 15 22 25 44 29 26	4
3 9 10 13 15 22 25 26 29 44	3
3 9 10 13 15 22 25 26 29 44	1

Total running time = 53

R-2.9

22 15 26 44 10 3 9 13 29 25	10
15 22 26 44 10 3 9 13 29 25	1
15 22 26 44 10 3 9 13 29 25	1
15 22 26 44 10 3 9 13 29 25	1
10 15 22 26 44 3 9 13 29 25	8 {4 swaps + 4 comparisons}
3 10 15 22 26 44 9 13 29 25	10
3 9 10 15 22 26 44 13 29 25	10
3 9 10 13 15 22 26 44 29 25	2
3 9 10 13 15 22 26 29 44 25	6
3 9 10 13 15 22 25 26 29 44	0

Total running time = 49

R-2.10

4 3 2 1	4 {4 insertions}
3 4 2 1	2 {1 comparison + 1 swap}
2 3 4 1	4 {2 comparisons + 2 swaps}
1 2 3 4	6

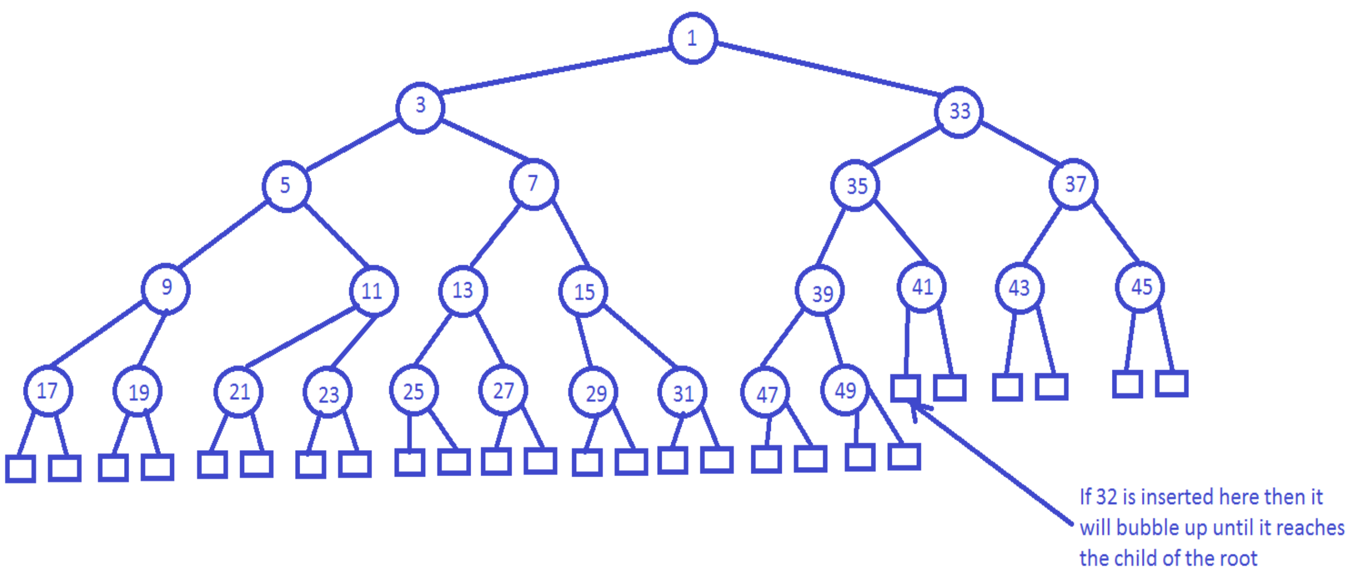
Total running time = 16

Any array sorted in descending order should have $\Omega(n^2)$ running time

R-2.13

Since the items in the vector are sorted, then the key of any item will be greater than or equal to its parent. As a consequence, tree T is a heap.

R-2-18



C-2.32

```
Algorithm reportSmallerKeys(H, x)
    s ← create new sequence
    i ← 1
    while  i <= H.size() ∧ H[i] <= x do
        s.insertLast(H[i])
        i ← i +1
return s
```