Some questions about priority queues and binary heaps

By *heap* we mean arrays representing binary min-heaps.

(a) An array with n elements can be turned into a heap in $\mathcal{O}(n)$.

(b) A min-heap becomes a max-heap by reversing the array.

(c) All elements at depth d in a heap are smaller than all elements at depth $d+1$.

(d) Insertion into a heap with n elements is $\mathcal{O}(\log(n))$ at worst case.

(e) An AVL tree can be used as a priority queue with the same runtime complexity as a heap.

(f) The nodes along a path from the root node to a leaf node of a heap are ordered from smallest to largest.

(g) If we reorder two sibling nodes of a heap, it remains a heap.

(h) A priority queue with constant time on all operations would give Dijkstra a runtime complexity of $\mathcal{O}(|V|+|E|)$.

(i) One can find the *largest* element in a *min*-heap in $\mathcal{O}(\log(n))$.

(j) *Over half* of the elements of a heap are located on the *two deepest* levels.