12.1-2

The binary-search-tree property guarantees that for any parent node, all nodes in the left subtree will be $\leq$ the parent and all nodes in the right subtree will be $\geq$ the parent. The min-heap property guarantees that for every parent node, both of its children will be $\geq$ the parent. While the binary-search-tree invariant guarantees that a tree always remain sorted left to right, the min-heap invariant only mandates that each ancestry chain remain sorted against itself, not taking into account other ancestry chains on different parts of the tree. Min-heaps cannot be traversed in $O(n)$ time because, unlike BSTs, they are not globally sorted. In order to sort the heap, one must run an algorithm that recursively:

1) moves the last node to the top spot
2) percolates the new top node down until it resides in a sorted position

Since step 1 takes $O(n)$ time and step 2 takes $O(\log n)$ time, heapsort is $O(n \log n)$ and thus cannot match the BST's $O(n)$ performance.