B) Time complexity: we can start with finding time complexity of each function. Time complexity of heapify function is O(logn), because this function performs comparison and swapping operations which have constant time complexity, also the function need to do these operations through the height of the heap, which is logn, so the time complexity is O(log n).
bottomUp function has O(n) time complexity and heapsort function has O(n log n)time complexity, because heapsort function creates heap by calling bottomUp function which is O(n) and it calls heapify function and does swapping for n-1 iterations , so the time complexity of heapsort is O(n log n)

C) Space complexity: this algorithm is doing in-place sorting which means it's not creating or using any extra space or data structure and all the operations are performed on the input array, so the code has O (1) or constant space complexity.

D) I guess the efficiency of the algorithm remains the same regardless of using max heap or min heap, because the only thing that changes is the comparison operation for doing swapping, other things and logic of the algorithm remains the same. The time complexity and space complexity also will remain the same, so the efficiency will remain same as well.