Problem 3

When we construct the heap, we can connect a ordering value for each element that we insert. (i.e. the first inserted item with order value 1, the sencond with order value 2, etc.). Then when we maintain the ordering property during the heapsort(actually when fixing-down), to let the algorithm to be stable, we just need to do a additional comparison based on their ordering value if the key of two items are the same.

The correctness and the running time of the algorithm is exactly same as the original heapsort. This is because we just add few comparison in some special steps. The correctness is based on the correctness of the heapsort. And for the correctness of the stable, the reason we add ordering value is to make sure the order of the elements that have the same key in output(new rearrange array) is the same as the original one. And the efficiency won't change since we add some constant time in some special steps of iteration.