17.1-1

Yes. Multipush runs in $O(n)$ time, and after using it the worst-case other operation would be Multipopping the k elements pushed by Multipop, another $O(n)$ operation. Taken together, this takes $2n = O(n)$ time, which amortized would result in $\frac{O(n)}{n} = O(1)$ time.

17.4-1

By doubling the hash table before its load factor approaches 1 too closely, we can avoid many iterations of the open-address hash function that occur when address collisions become more common. If we modify the insert function to table double when the load factor reaches a certain level, we can avoid the near-linear inserts that occur when you insert into an almost-full open address table. If you double around a load factor of 0.5, you have done n operations at $O(1)$ to get to that point, then you do a single $O(n)$ operation of table doubling and copying, for an amortized cost of $O(n)/n = O(1)$.