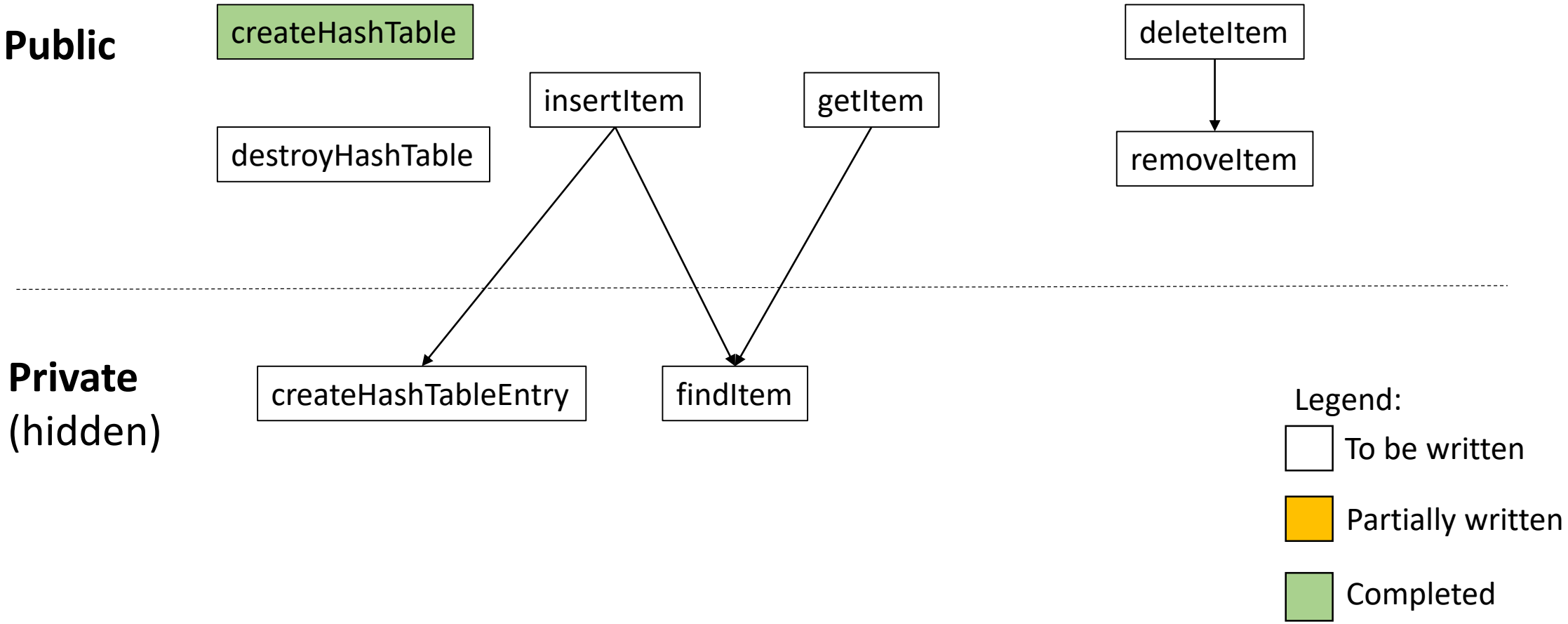
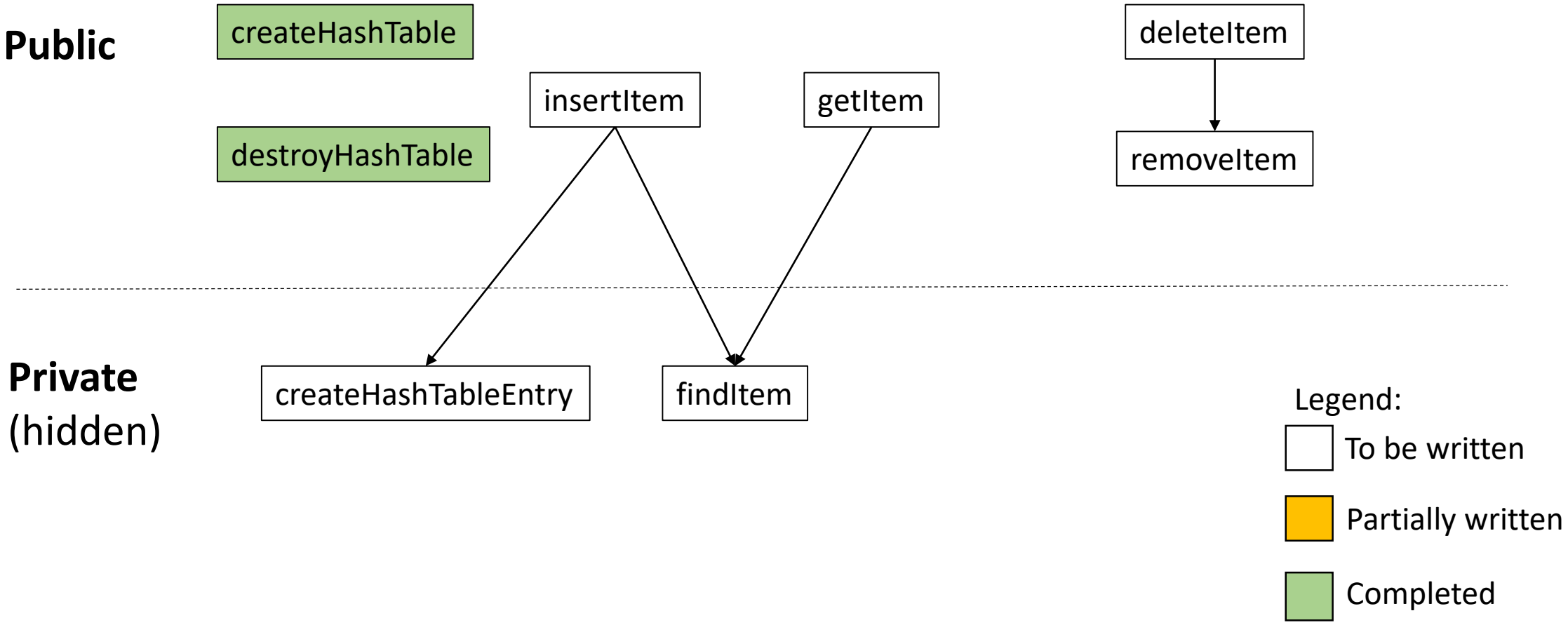
P2-1: Who Calls What



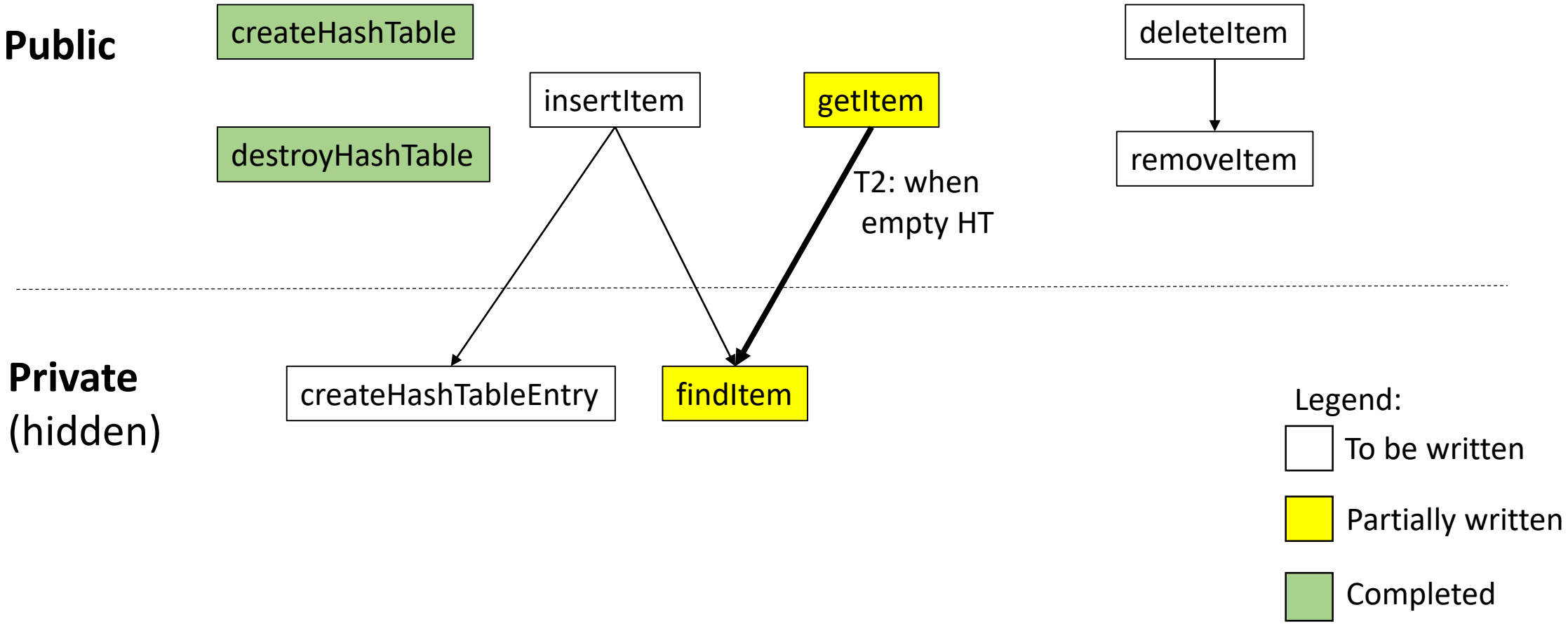
Test 1: CreateDestroyHashTable

Create/destroy HT used by all test cases. Implement destroyHashTable (see hashtable.h for specification).



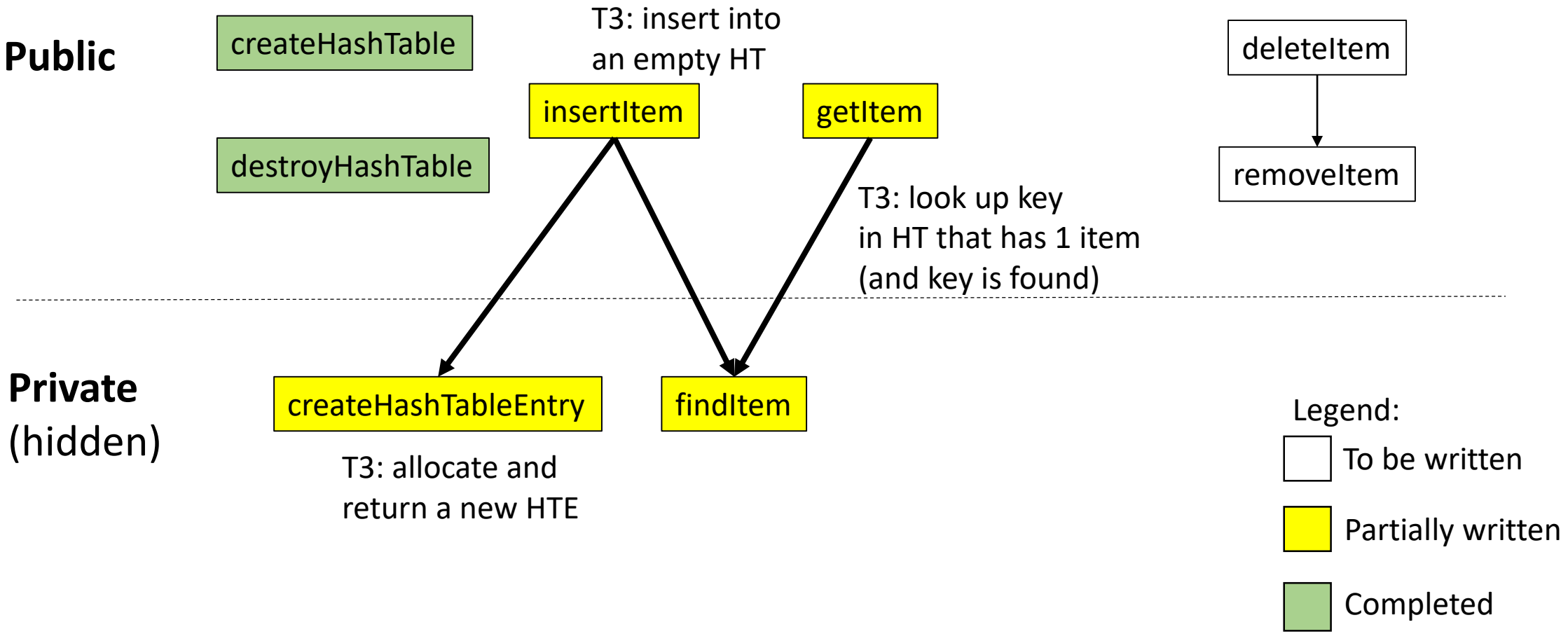
Test 2: GetKey_TableEmpty

Goal: implement enough of findItem and getItem to make this test succeed when the hashtable is empty. Look at the specification for each. What is the difference in what they return?



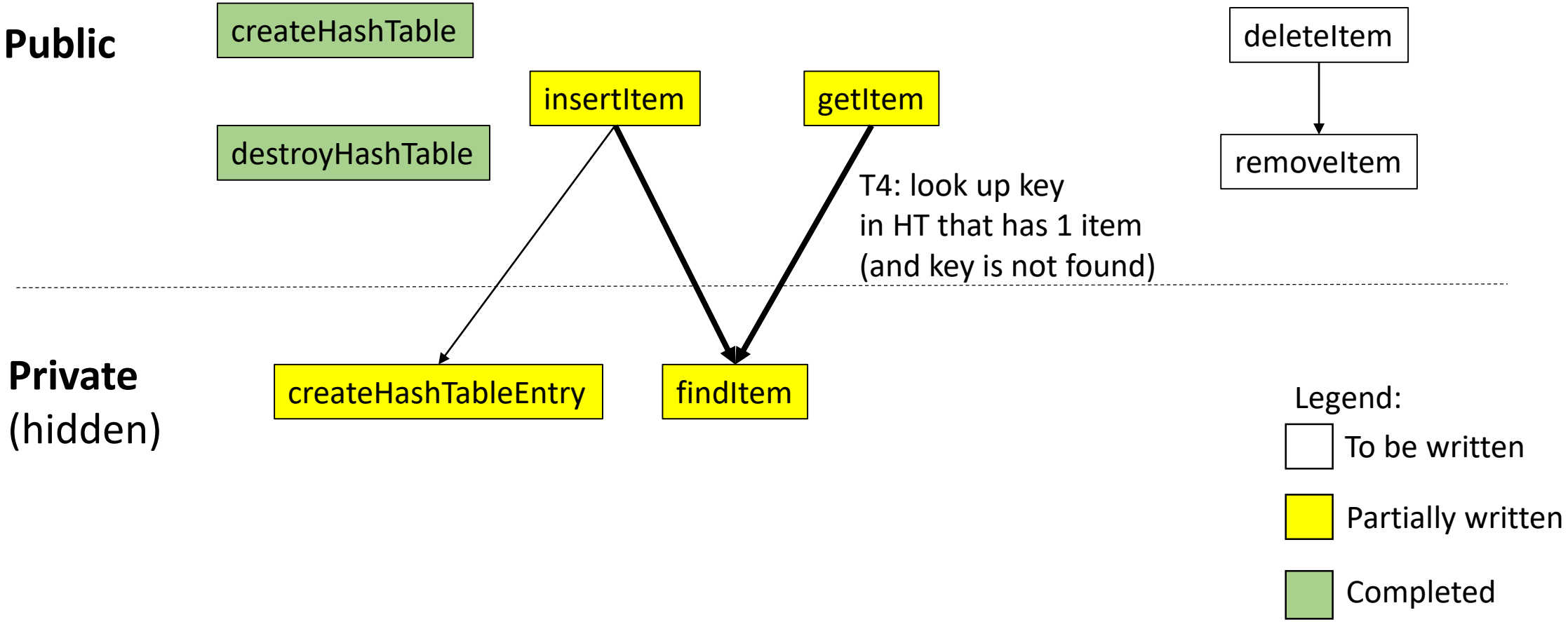
Test 3: GetSingleKey

Goal: implement createHashTableEntry and enough of insertItem, findItem, and getItem, to make this test succeed. Insert one item into HT and look up key that matches the key of item inserted. Note: both insertItem and findItem must apply the HT's hash function to the key to get a bucket number. Use the bucket number to index into the HT buckets to get the head of the bucket list where the item belongs.



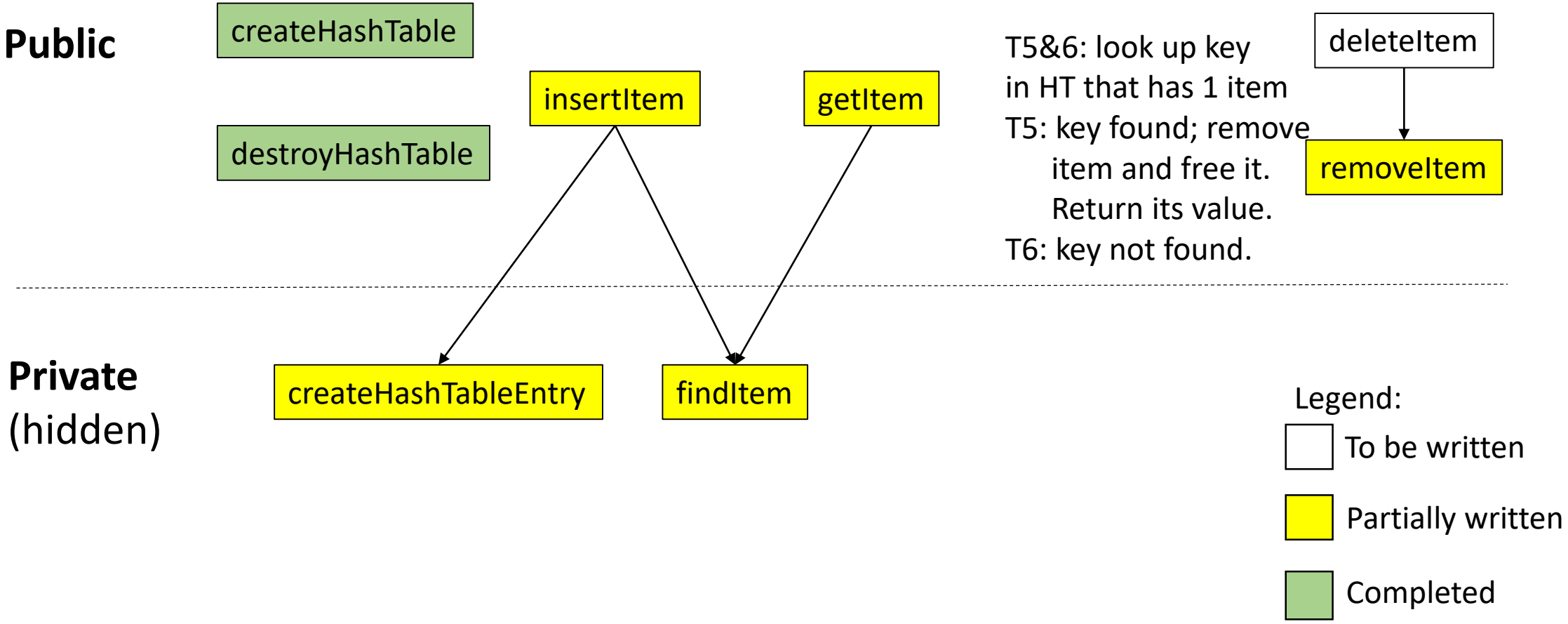
Test 4: GetKey_KeyNotPresent

Goal: Similar to Test 3, but look up key that does *not* match key of item inserted.



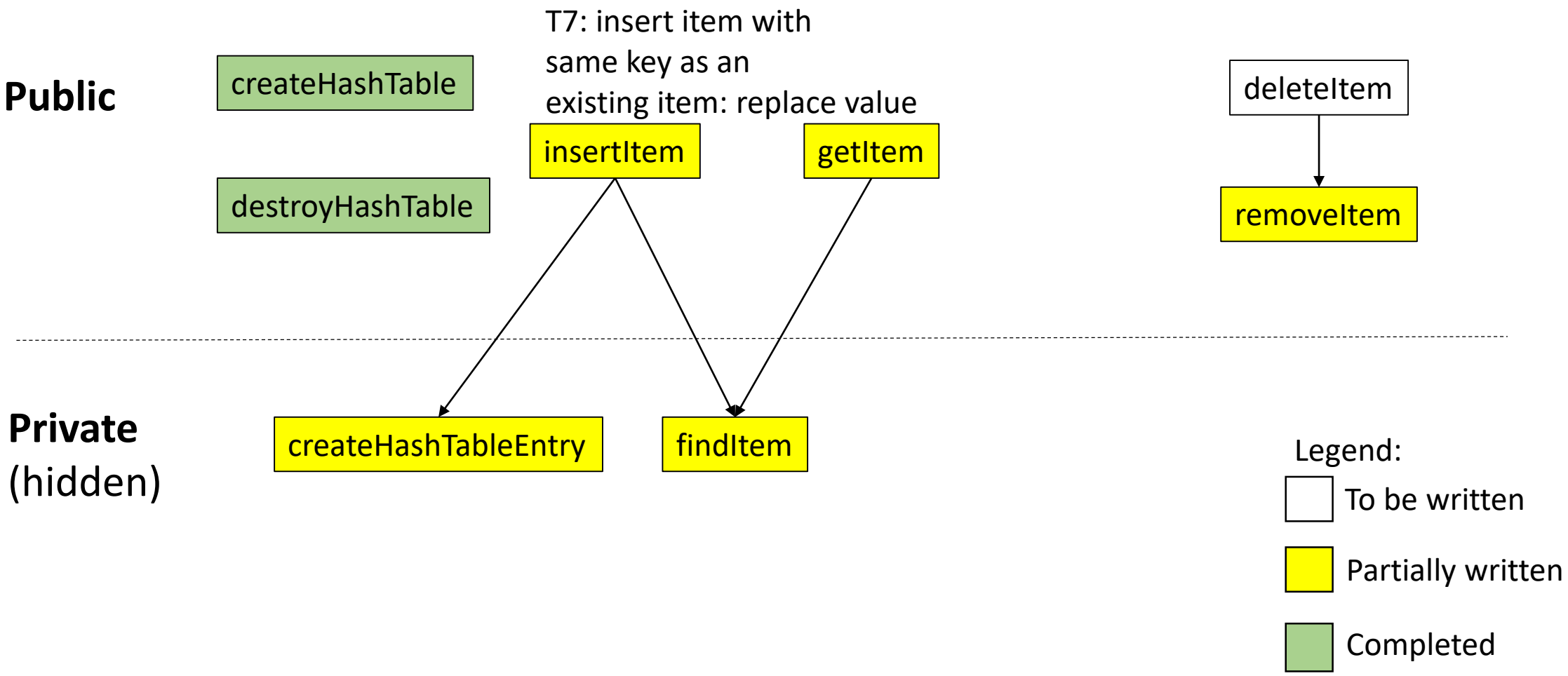
Test 5 & 6: SingleValidRemove & SingleInValidRemove

Goal: Implement enough of removeItem to be able to remove an item from a HT that has 1 item in it and that item's key matches the key given as input. Tests insert 1 item into an empty HT and T5 removes the same item while T6 tries to remove an item w/ a different key (not present in the HT).



Test 7: InsertAsOverwrite

Goal: implement more of insertItem, findItem, and getItem, to make this test succeed. Insert an item with a key that matches the key of an item already in the hashtable.



Continue to add tests and complete the functions

For example:

- Insert into nonempty HT and test access of items
  - 2 items in different buckets
  - 2 items in same bucket (different keys, but both key hash to same bucket)
  - >2 items in same bucket
- Remove and delete – what are edge cases? (e.g., removing from head, from middle…)