**Problem 7**

Compare the tombstone and lock-and-key methods of avoiding dangling pointers and memory leakage, from the points of view of safety and implementation cost.

Answer:

Essentially tombstones end up being costly in both time and space. The tombstone method does not allow their own deallocation. This will mean that any other heap access will need another level of indirection; thus causing another machine cycle for virtually all computers. This does provide a good amount of extra safety, but the cost does not seem worth it to many developers.

Lock and key pointer values are ordered pairs. It is a key and address. Heap-dynamic variables are represented as the storage for the variable along with a header cell that stores an integer ‘lock’ value. When one of these variables is allocated, the lock value is created. When a heap dynamic variable is allocated, the lock value is then created and placed in the lock cell of the heap dynamic. Then, every access to the dereferenced pointer compares the key value of the pointer to the lock value in the heap-dynamic variable. If they match, the access is legal; otherwise the access is treated as a run-time error. Any copies of the pointer value to other pointers must copy the key value. If after that, something changes, then the key and value wont match which will not allow a legal access. Therefore it is still safe.

Source:

https://quizlet.com/287740641/flashcards