

CSE 332: Data Structures and Parallelism

Exercises (Hashing)

Directions: *Submit your solutions on Gradescope. You must submit a pdf file.*

EX09. Hshg (20 points)

In this problem, you will think about how lazy deletion is handled in open addressing hash tables. Refer to [slide 13 on Oct 22](#) for a description of lazy deletion. You should NOT assume that any changes other than those specifically stated are made to the hashtable (e.g. operations still work exactly the same way as we discussed in lecture).

For both part a) and part b) we are asking you to compare the proposed modification to how an open addressing hash table with lazy deletion would normally work.

We have an open addressing hash table that uses lazy deletion. The cell *X* is marked as “deleted”. Consider the following two proposed modifications to how the **find** operation works:

(a) [10 Points] Proposed modification A:

- While probing, a successful find operation hits and moves past cell *X* and finds the key it is looking for in cell *Y*.
- The find operation then moves the found key to cell *X*, marks cell *X* as “no longer deleted”, and **marks cell *Y* as “open” (as if it had never had a value in it)**.
- The find operation then returns the found value.

If this proposed modification is used, would the resulting hash table work better or worse than a normal open addressing hash table using lazy deletion. **Explain your answer.**

(b) [10 Points] Proposed modification B:

- While probing, a successful find operation hits and moves past cell *X* and finds the key it is looking for in cell *Y*.
- The find operation then moves the found key to cell *X*, marks cell *X* as “no longer deleted”, and **marks cell *Y* as “deleted” (it contains no value, but we treat it as a collision)**.
- The find operation then returns the found value.

If this proposed modification is used, would the resulting hash table work better or worse than a normal open addressing hash table using lazy deletion. **Explain your answer.**