Closed Hashing

- We have presented *open* hashing, where all items that hash to the same position are stored in a bucket that is represented by a linked list.

- With *closed* hashing, whenever an item is hashed to a position that is already occupied, search for some other place in the table to put it.

- So all buckets have $\leq 1$ element, and items (or pointers to them) are stored directly in the table.

- Although it is possible to implement DELETE, closed hashing is usually used when only INSERT and MEMBER operations are needed.

- Common methods for searching for an empty bucket:
  
  * **Linear probing**: Scan from that position forward (wrapping around if you get to the end of the table) until an empty position is found.
  
  * **Quadratic probing**: Scan forward by quadratic amounts (in the hope of avoiding bunching up of entries).
Note:

- Quadratic probing from a position $i$ in a hash table of buckets 0 through $m-1$ can work by searching buckets $i + j^2 \text{ MOD } m$ and $i - j^2 \text{ MOD } m$, for $1 \leq j \leq (m-1)/2$.

- For the case that $m$ is a prime of the form $4k+3$ for some integer $k$, all positions of the hash table get visited.

- In general, for any probe function, keep probing until you find an empty slot or until you get back to where you started (and return an error).

- If too many probes are being used, it may be time to move to a larger hash table ...