Hashing

5/20/26

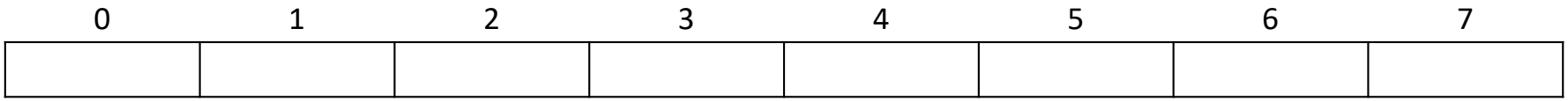
Administrivia

- 2 new assignments coming
  - HW 8 on radix sort and hashing
  - HW 9 on creating your own design problem
- I'm at a conference next week
  - No class Monday and no lab Thursday
  - Zoom class on Wednesday (just questions and doing your design problems)
- Final exam: Monday 6/1 1:30-4:30pm in our room
  - Open notes, written, you need to stay in the room
  - comprehensive but weighted toward the last part of the course

Back to Set and Map

- Need to quickly find a key (and associated value)
- Main operations are find, insert, delete

A different approach: Hashing

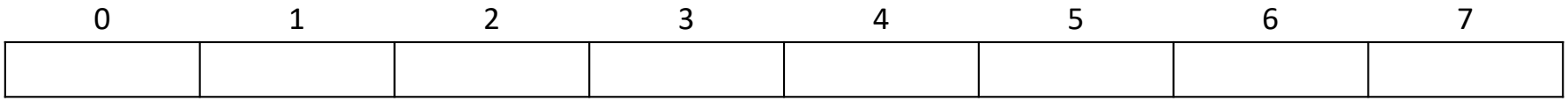


- Based on an array
  - Use hash function to convert key to integer
  - Use % to convert it to an array index

- Closed addressing:
  Each entry is a list of keys mapping to that cell

Where does 12 go in the following hash table?

(Assuming identity hash function)



A. 0

B. 2

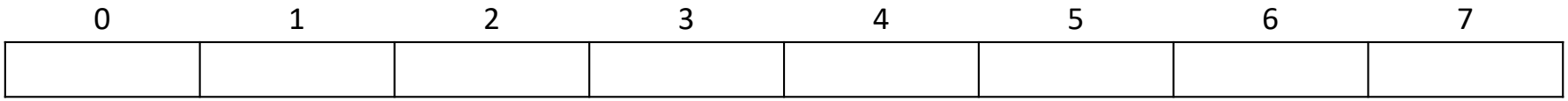
C. 4

D. 6

E. None of the above

Where does 12 go in the following hash table?

(Assuming identity hash function)



A. 0

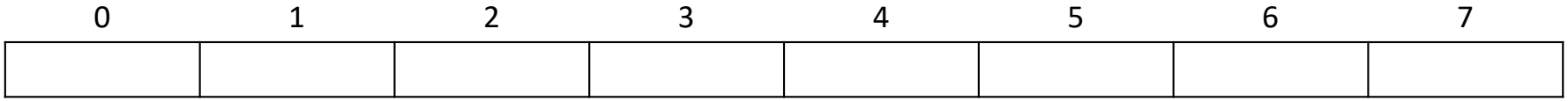
B. 2

C. 4

D. 6

E. None of the above

Alternate: Open addressing



- Only one key can be in each array cell
- When a second key tries to join, find a different place
  - Linear probing
  - Quadratic probing
  - Double hashing

Tricky point: Removing a key

- Can’t just remove it or other keys might disappear from the table
- Instead, replace with a “bridge” that has no key but represents a full cell

Consider an initially empty hash table of size 7 implemented with open addressing and linear probing. Draw a diagram showing its contents after each operation in the following sequence (using the value of each number as the value of its hash function):

add(10), add(3), add(4), add(17), delete(4)

What ADTs would you use for this problem?

A. List or Priority Queue
B. Queue or Stack
C. Map
D. Set
E. More than one of the above