Abstract Data Types: Hash Tables

Hash Table
Implementation of a Set 1
Hash Tables

A hash table stores \((key, value)\) pairs, using a hash function to map a key into a table. Key challenges are: a) dealing with hash collisions, and b) dealing with load (how big to make the table).

Two broad approaches:

- **Separate chaining**
  - Hash table entries are lists. \((key, values)\) are in lists.

- **Open addressing**
  - Hash table entries are \((key, value)\) pairs.
  - Collisions resolved by *probing* – e.g. find next empty slot
Abstract Data Types: Hash Table

fruit

- apple
- orange
- banana
- pear
- apricot
- peach
- mango
- plum
- cherry
- grape
fruit.add("apple")
fruit.add("orange")
fruit.add("banana")
fruit.add("pear")
fruit.add("apricot")
fruit.contains("orange")

fruit.contains("mango")

fruit.contains("fig")

fruit.contains("fig")
Abstract Data Types: Hash Table

- apple
- banana
- apricot
- peach
- mango
- pear
- plum
- grape
- cherry

- a-f
- g-m
- n-t
- u-z

fruit
Abstract Data Types: Hash Table

![Diagram illustrating hash table with fruits as keys and their order within the table: apple, banana, apricot, cherry, mango, grape, orange, pear, peach, plum. The diagram shows the structure of the hash table with keys grouped by their first letters: a-f, g-m, n-t, u-z.](image-url)
Abstract Data Types: Hash Table

fruit.contains("grape")

fruits

- a-f
- g-m
- n-t
- u-z

- apple
- banana
- apricot
- cherry
- mango
- grape
- orange
- pear
- peach
- plum