Hash Tables: Takeaways

Syntax

- Retrieving an unicode character code:

```
ord(string)
```

- Creating a simple hash function:

```
def simple_hash(key):
    key = str(key)
    return ord(key[0])
```

- Implementing a hash table:

```
class HashTable():
def __init__(self, size):
    self.array = np.zeros(size, dtype=np.object)
    self.size = size
def __getitem__(self, key):
    ind = hash(key) % self.size
    for k,v in self.array[ind]:
        if key == k:
            return v
def __setitem__(self, key, value):
    ind = hash(key) % self.size
    if not isinstance(self.array[ind], list):
        self.array[ind] = []
    replace = None
    for i,data in enumerate(self.array[ind]):
        if data[0] == key:
            replace = i
    if replace is None:
        self.array[ind].append((key,value))
    else:
        self.array[ind][replace] = (key, value)
```

- Returning the hash value of an object:

```
hash(object)
```

Concepts

- A hash table allows us to store data with a key that is associated with a value.
- Time complexity used to look up a value when using a key is .
- A hash table stores their data in an array.
- A modulo, denoted with "%" in Python, is a mathematical operator that finds the remainder after a number is divided by another. For example:
    - 11 % 10 = 1
    - 10 % 10 = 0
    - 20 % 3 = 2
    - 23 % 9 = 5

- A hash collision occurs when two different values results in the same hash.
- One way to avoid collisions is to store a list of values at each array position.
- The time complexity is when all the elements in the hash table are in a single list.
- The Python `dict` class is implemented to optimize for speed, but the following principles apply:
    - Insertion is generally , but worst-case can be .
    - Indexing is generally , but worst-case can be .
- A dictionary is an excellent data structure since hash tables can be combined fairly easily. It's very commonly used when doing distributed computation, and understanding the memory and lookup time constraints can be very helpful.

Resources

- Unicode HOWTO
- Python dictionary implementation