1. What internal data structure does a Python dictionary use that enables it to provide a key-value data store behavior with O(1) insert, delete, and search?

Python uses a dictionary which is a Key-Value data structure with behavior O(1) inset, delete, and search.

1. What is the algorithmic complexity of searching for a particular *value*, not a particular key, in a dictionary?

A Hash map which is an abstraction that uses and array and hashing function in order to store and retrieve values.

1. How do hash tables handle key collisions?

Hash tables handle key collisions through a process called collision resolution and there are two main collision resolutions: One of which is called rehashing, which looks into the hash table sequentially through the slots and tries to find an open slot that can hold the key that caused the collision. The other is called chaining and that’s when it takes the value and creates a linked list within that key.

This process is called **collision resolution**.