Sort complexity is nlogN

Stack. Last In First Out. Used for depth-first search (DFS).

Queue. FiFo

HashTabes. Use hashing. Hash functions are measured based on their load factor, with a load factor of 1 being the worst. In the worst-case scenario, all hashes are the same and are stored in one entry which takes o(n) to search. In average cases, with a good hash function, it is o(1).

Methods for avoiding collision: Separate chaining, open addressing

Python implements a hashtable as a dict

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

Graphs

Traversals: DFS and BFS

DFS uses Stack (or recursion which uses the internal call stack). Acts on the end of a list. Moves in one direction till there is no more edge before it moves to another node.

BFS uses Queue (cannot use recursion because recursion uses internal stack). Acts at the beginning of a list. Explores edges in a level before moving on to the next level (or depth). Useful for finding shortest distance, as it does not travel endlessly in one way before retracing.