Optimisation in Python

Cpython is the interpreter that is considered to low level language like a C and CPP  
=>use time method

=>Cprofiling=>makes the program faster to run

=>LineProfiler=> helps to know bottleneck of CPUs and RAM usage in our code

=>bisect means to implement binary search

=>Lists are dynamic arrays and tuples are static arrays.Lists are not cached but tuple are cached at runtime

=>Dictionary is a key-value pair and set is a collection of unique key

=>hash Table

=>Other ways to write efficiently  
1. Iterate with enumerate instead of range link  
enumerate uses both the index and the data stored at that index so we can directly check the data

2.Use list Comprehensions Instead of for raw loops

Squares=[]

for i in range(10):

Squares.append(i\*i)

print(squares)

squares=[i\*i for i in range(10)]

print(squares)

3.Sort complex iterables with sorted()

4.store unique elements with sets

5.safe Memory with Generators instead of using too large list because it only generates one item at a time

6.Define default values in dectionaries with .get() and setdefault()

7.count hashtable objects with collections.counter

8.Format strings with f-strings

9.strings are immutable elements so if we use + operator new string will be created each time therefore we should use join()

10. Merge dictionaries with double asterisks used in python 3.5+

11.simplify if statement for multiple checks

Use like

colours=[“red”,”green”,”blue”]

c=”red”

if c in colors:

print(“is main color”)