Constant Time Complexity: Takeaways ↗

Syntax

- Concatenating all words in a list:

```
concat = ''.join(word_list)
```

Concepts

- A constant time complexity algorithm is an algorithm whose execution time does not depend on the input.

- Constant time algorithms can take a long time to run, but since their execution time does not grow as data increases, with large enough data, they always become more efficient than non-constant time algorithms.

- Line execution count only provides accurate time complexities when there are no function calls.

- To analyze the time complexity of an algorithm, we analyze the time complexity of every line of code and add them together in the same we as we did with line execution counts.

- A computer can allocate single memory locations of continuous ranges of memory locations. Any allocated memory location can be accessed in constant time.

- As long as the numbers do not grow arbitrarily large, basic arithmetic operations addition, subtraction, multiplication, and division are computed in constant time.

- We should only use lists when the operations that we need to use on the data are limited to `list.append()` , `list.pop()` , and looping over all list elements.

Resources

- [Python time complexities of lists, sets, and dictionaries](#)