IMPERIAL

BIOE50010 – Programming 2

Computer Lab 7: Algorithms

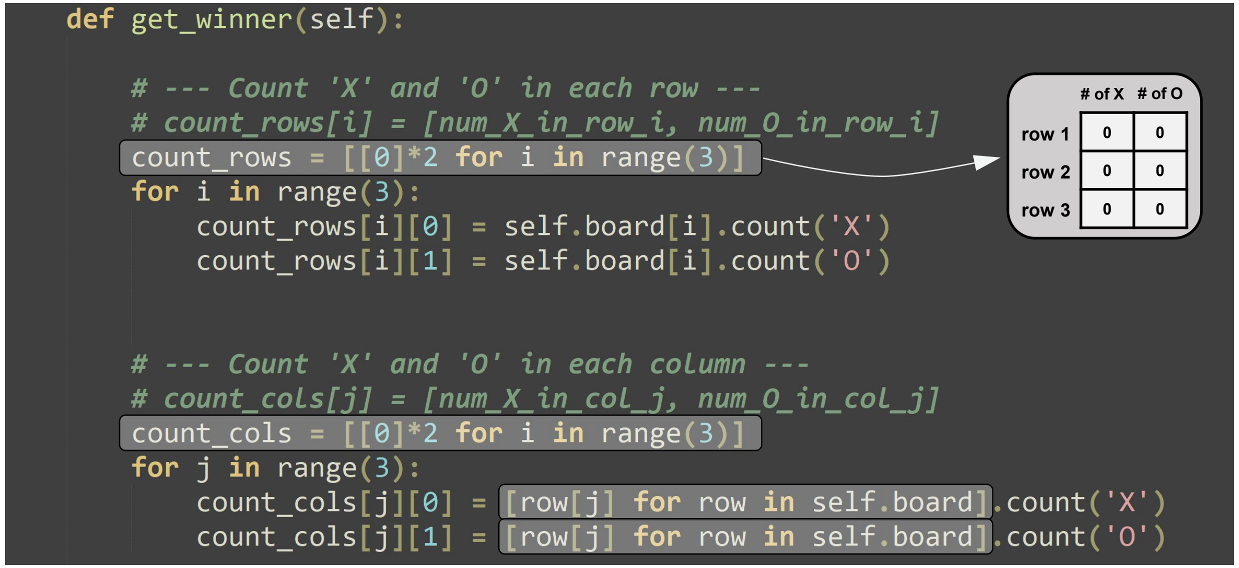
Binghuan Li, Maria Portela, Gauthier Boeshertz, Samuel George-White,
Yilin Sun, Kamrul Hasan, Wenhao Ding, Siyu Mu, Lito Chatzidavari

16 November, 2025

Feedback

- No cohort-level issue identified in week 6.
- However, any individual questions, speak to a TA.

- I would suggest you prioritise **Labs 04-06** if your have yet finished.
  - Compare your solution to the suggested solution.
  - There exists many approaches to solve the same problem.
  - If you see a new syntax that you do not understand, talk to us.



```python
def get_winner(self):

    # --- Count 'X' and 'O' in each row ---
    # count_rows[i] = [num_X_in_row_i, num_O_in_row_i]
    count_rows = [[0]*2 for i in range(3)]
    for i in range(3):
        count_rows[i][0] = self.board[i].count('X')
        count_rows[i][1] = self.board[i].count('O')


    # --- Count 'X' and 'O' in each column ---
    # count_cols[j] = [num_X_in_col_j, num_O_in_col_j]
    count_cols = [[0]*2 for i in range(3)]
    for j in range(3):
        count_cols[j][0] = [row[j] for row in self.board].count('X')
        count_cols[j][1] = [row[j] for row in self.board].count('O')
```

	# of X	# of O
row 1	0	0
row 2	0	0
row 3	0	0

Suggested solution to Lab 05 (`board.py`): extensive use of **list comprehension** in **`get_winner()`**.

Although this concise style can reduce readability and isn't generally recommended, you should still be familiar with how it works. I have added explanatory annotations to the solution file.

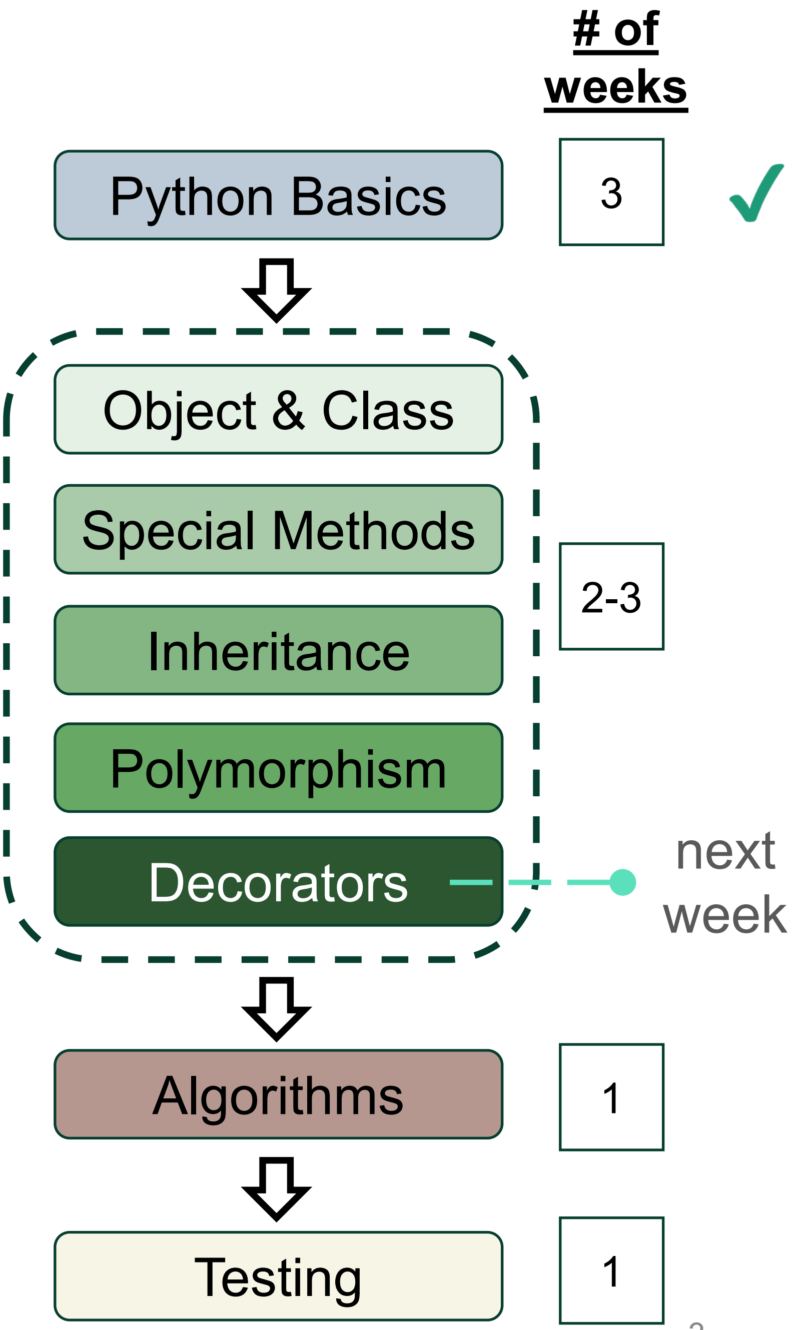
Progress Check

Revision Points (from weeks 4-6)

- Syntex for **class definition** and **object instantiation**.
- Overload operators using **special methods**.
- Use **inheritance** to design a programme, access attributes and methods from the super class.

Questions outside the classroom? ed discussion

Week 7:
we are here



Your task today

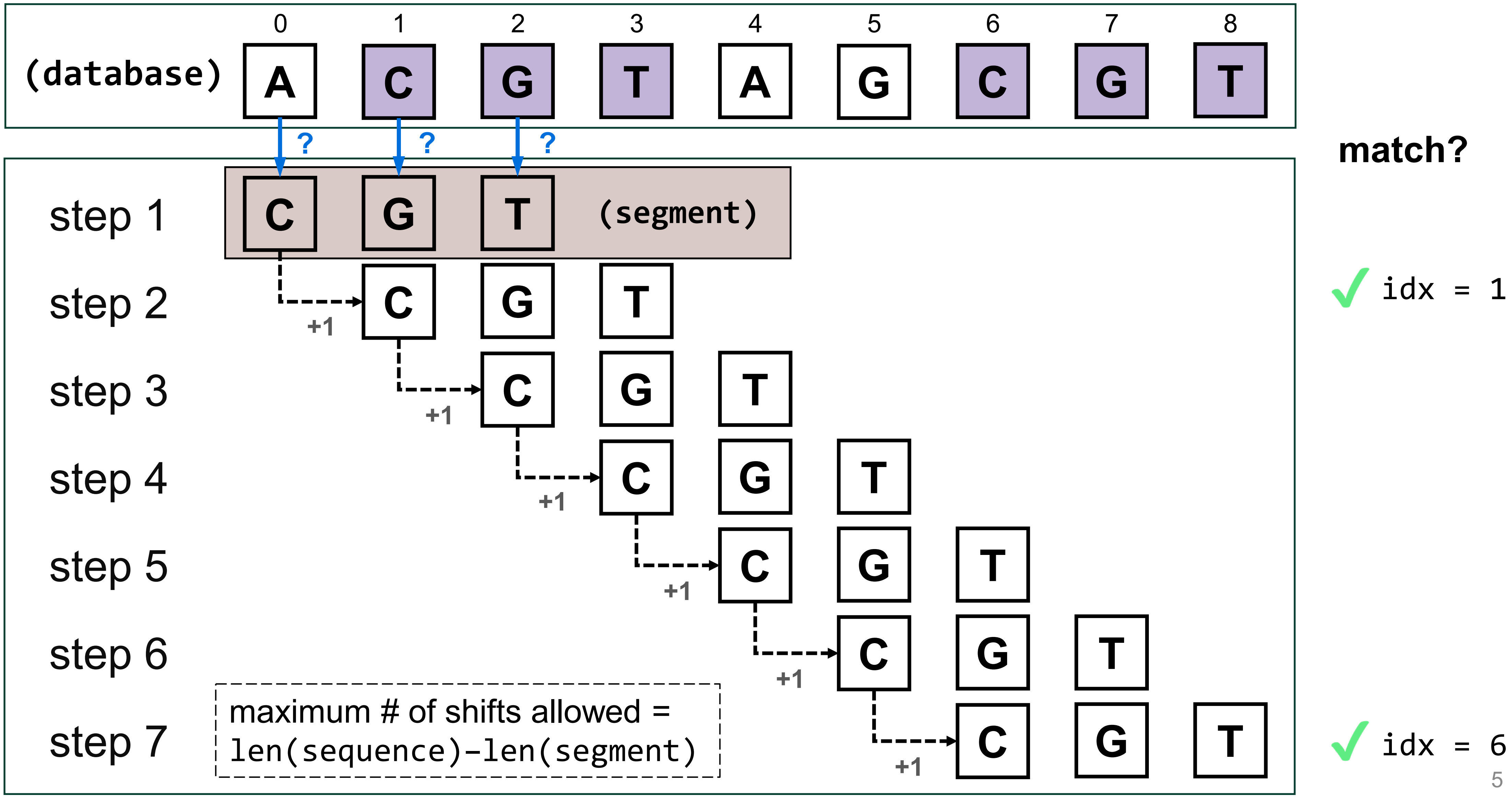
Implement a `find()` method to the class `Dna` to search a user-defined **DNA segment** within the **DNA database**.

Return the **starting index** of the nucleotide within the DNA database.

To start…

- Study the suggested solution of Lab 06 (`dna_oop.py`) from Blackboard. Make sure you are confident about the logics.
- Read and study the driver script and the sample output from the lab sheet carefully.
- You may assume the length of the DNA segment is shorter than the DNA database – no need to consider overflow/paddings.
- Once you have completed – can you optimize and accelerate your code?

One Possible Matching Algorithm



	0	1	2	3	4	5	6	7	8
(database)	A	C	G	T	A	G	C	G	T

? ? ?

match?

step 1 C G T (segment)

step 2 +1 C G T ✓ idx = 1

step 3 +1 C G T

step 4 +1 C G T

step 5 +1 C G T

step 6 +1 C G T

step 7 +1 C G T ✓ idx = 6

maximum # of shifts allowed =
`len(sequence)-len(segment)`