Lab 5: Search & Sort

SUNY Korea - Francois Rameau

Spring 2023

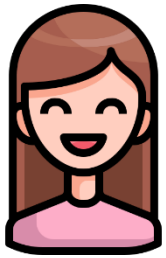
GitHub Classroom

Lab 5 – Search and sort

Finding love with Python

Today you are an entrepreneur, you just opened your online dating platform

Your role is to find the ideal partner to our two clients

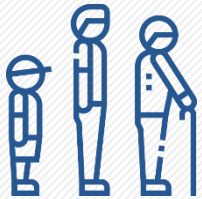


Pamela

George

Our database

For each user you stored the following

Name	Age	Height	Income
	 In year	 In cm	 In k$

Each person is stored in a list

```
user = ["Joe", 25, 175, 50]
```

Name Age Height income

Our database

Let's assume, you have already collected data from male and female that you saved as CSV files (comma-separated values)

list_females.csv

```
Peyton,26,157,49
Leah,27,163,54
Audrey,23,156,50
Savannah,24,168,57
Alyssa,29,165,59
Aubrey,22,160,52
Kylie,31,162,61
Bella,25,158,55
Claire,26,157,50
Stella,30,171,61
Maya,24,163,56
Gabriella,28,167,60
Skylar,21,155,48
Samantha,23,162,55
Allison,26,160,52
Caroline,30,169,61
Elena,25,158,55
Isabelle,27,163,54
Ariana,32,154,57
Genesis,20,172,47
Ruby,26,157,49
Reagan,31,162,61
Clara,22,160,52
Naomi,23,156,50
Khloe,29,165,59
Annabelle,24,168,57
Alexa,26,157,50
```

list_males.csv

```
Samuel, 27, 168, 40
Derek, 34, 175, 52
Frank, 40, 188, 70
Tyler, 28, 172, 48
Raymond, 37, 180, 65
Shawn, 35, 178, 60
Travis, 26, 170, 42
Gregory, 38, 183, 67
Jeffery, 33, 174, 52
Trevor, 32, 174, 52
Douglas, 39, 183, 67
Austin, 29, 169, 45
Lee, 31, 173, 48
Cody, 27, 169, 45
Bryan, 37, 180, 65
Luis, 33, 176, 52
Carlos, 35, 178, 60
Alan, 38, 183, 67
Mario, 42, 188, 70
Juan, 32, 174, 52
Jeremiah, 26, 171, 45
Scott, 34, 175, 52
Alexander, 29, 171, 45
Victor, 36, 180, 65
Joel, 31, 173, 48
Jeff, 40, 187, 75
Johnny, 28, 172, 30
```

1. Open the database

A function to read the database is already implemented

```
candidates_list = read_database('list_males.csv')
candidates_list = candidates_list[1:]
```

Just to remove the header

The database is stored in a nested list

```
[['Douglas', 39, 183, 67],
['Austin', 29, 169, 45],
['Lee', 31, 173, 48],
['Cody', 27, 169, 45],
['Bryan', 37, 180, 65],
['Luis', 33, 176, 52],
['Carlos', 35, 178, 60],
['Alan', 38, 183, 67],
['Mario', 42, 188, 70],
['Juan', 32, 174, 52]]
```

```
candidates_list[2]
```

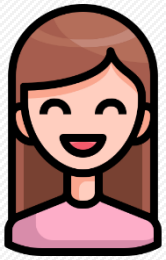
2. The clients

The goal of this lab is to find a good match to our two clients

We will define a good match by "a partner with similar features"

Close age, close height and close income

For instance



Pamela

`[25, 157, 65]`

Henry

`[65, 190, 55]`

Peter

`[24, 175, 60]`

Peter has a closer profile than Henry then he is a better match for Pamela

3. Euclidean distance

TO DO:

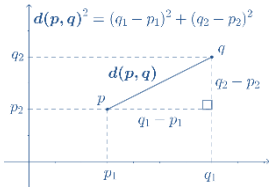
Implement a function to compute the Euclidean distance between a client (Pamela or George) and one person in the dataset

Euclidean distance

$$d(p,q) = \sqrt{\sum_{i=1}^{N}(p_i - q_i)^2}$$

In our case

$$d(person1, person2) = \sqrt{(age1 - age2)^2 + (height1 - height2)^2 + (income1 - income2)^2}$$



A small distance represent a good match!
Try to implement it with a **FOR** loop in the function

`EuclideanDistance()`

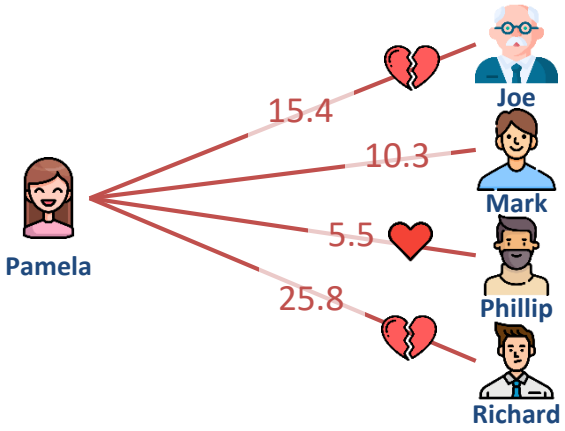
4. Compute the distance between the client and the candidates

TO DO:

Compute all the distances between Pamela (or George) and all the candidates in the database (male or female)

You will implement this in `ComputeAllDistances()`

→ This function will return the list of distances between one client (Pamela or George) and a database.

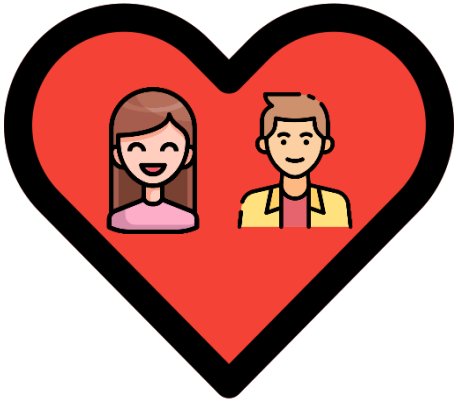


5. Find the best match

TO DO:

Complete the function `FindMin()` that returns the minimum value in a list and its corresponding index

Using the index returned by your function, display the name and characteristic of the best match for our client



By the way, you just implemented a naïve nearest neighbor search

6. Bonus

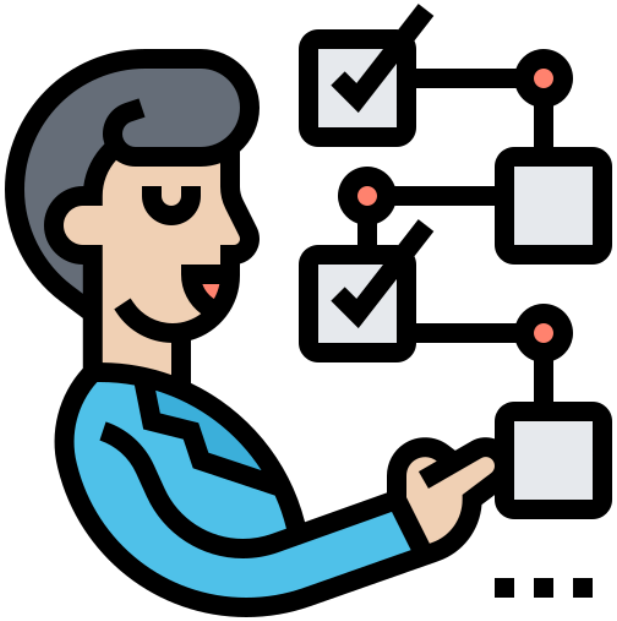
- What can you say about the complexity of this search strategy?
- How would you return the top 5 best match? (try to implement it)
- Can we return all the results given a range over each criterion (no need to implement)


FAIL
PASS
Novelty today
THE UNIT TEST!

Unit test

! Terminology alert !

A **unit test** is a type of software testing technique that focuses on testing individual components or functions of a program in isolation to ensure they perform as expected. This helps developers **identify and fix errors** early in the development process.



Unit test

Note that you now have:

YourHomework.py

AND

YourHomework_test.py ⟸ This file contains the unit tests

Do not touch

Do not modify the unit test by any mean

Unit test

Only work on the origin *.py file I am providing you!

Unit test

- Does it change the way I push my work on Github?
  - Not at all, push and commit as before, **follow Lab0 very carefully**
- Why do we do unit test?
  - Because it will test every single piece of your code to make sure it is working as expected. It saves time for the TA and ensure a fair evaluation
- What unit test does for me?
  - It will directly tell you if you are code is right or wrong such that you can correct a function that you have not coded properly!

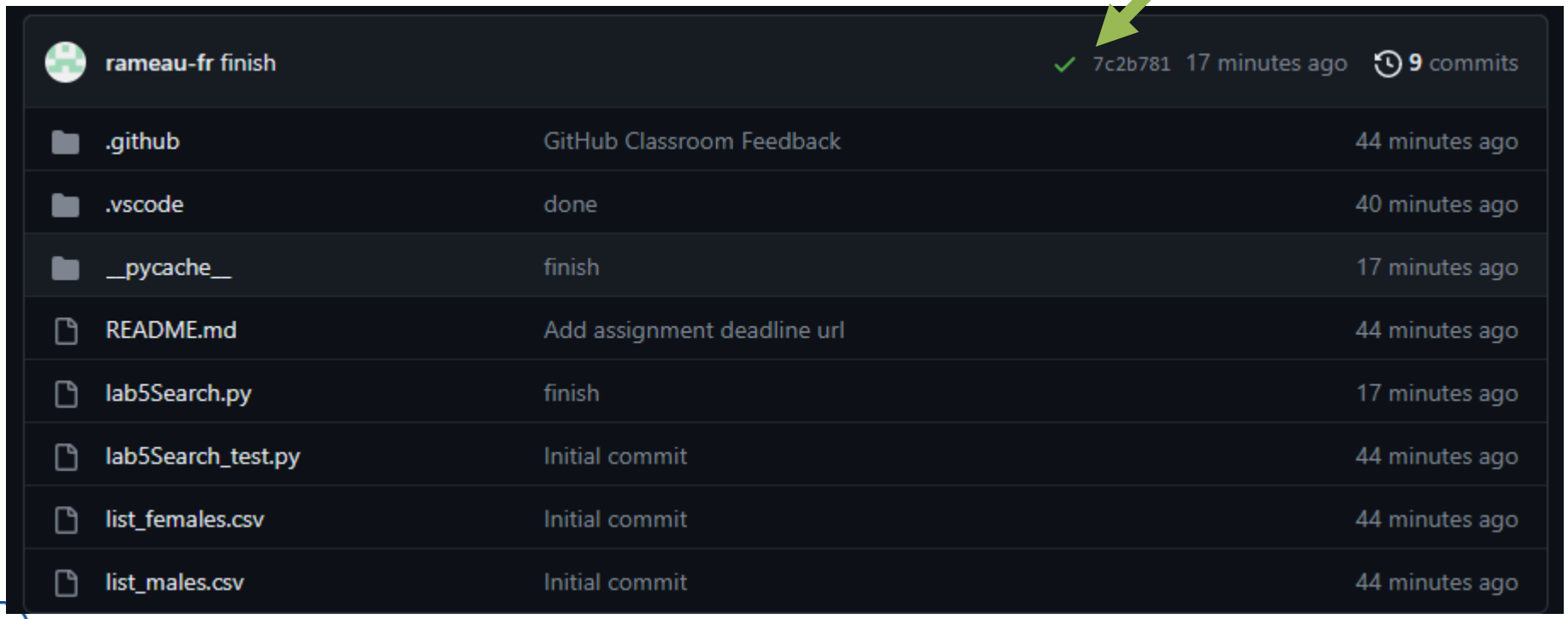
Unit test

Let's push a code on GitHub to see what is happening!!

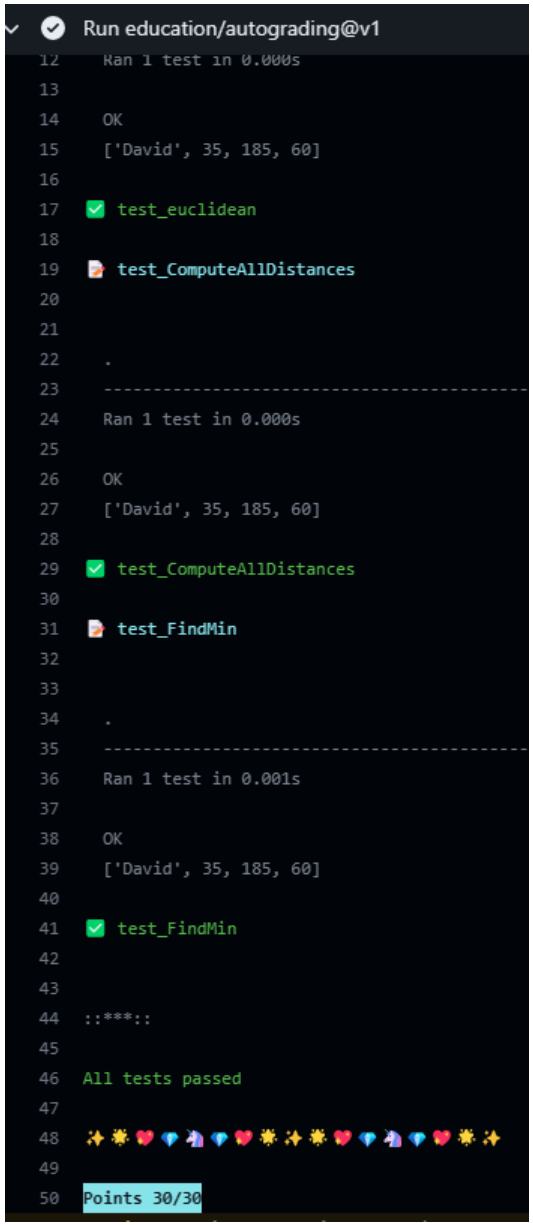
Two possibilities

1.Your code is working as intended

You have this little green check that means that all your code is working as intended
You can click on it!



Unit test



You will directly see which tests has been successful or failing

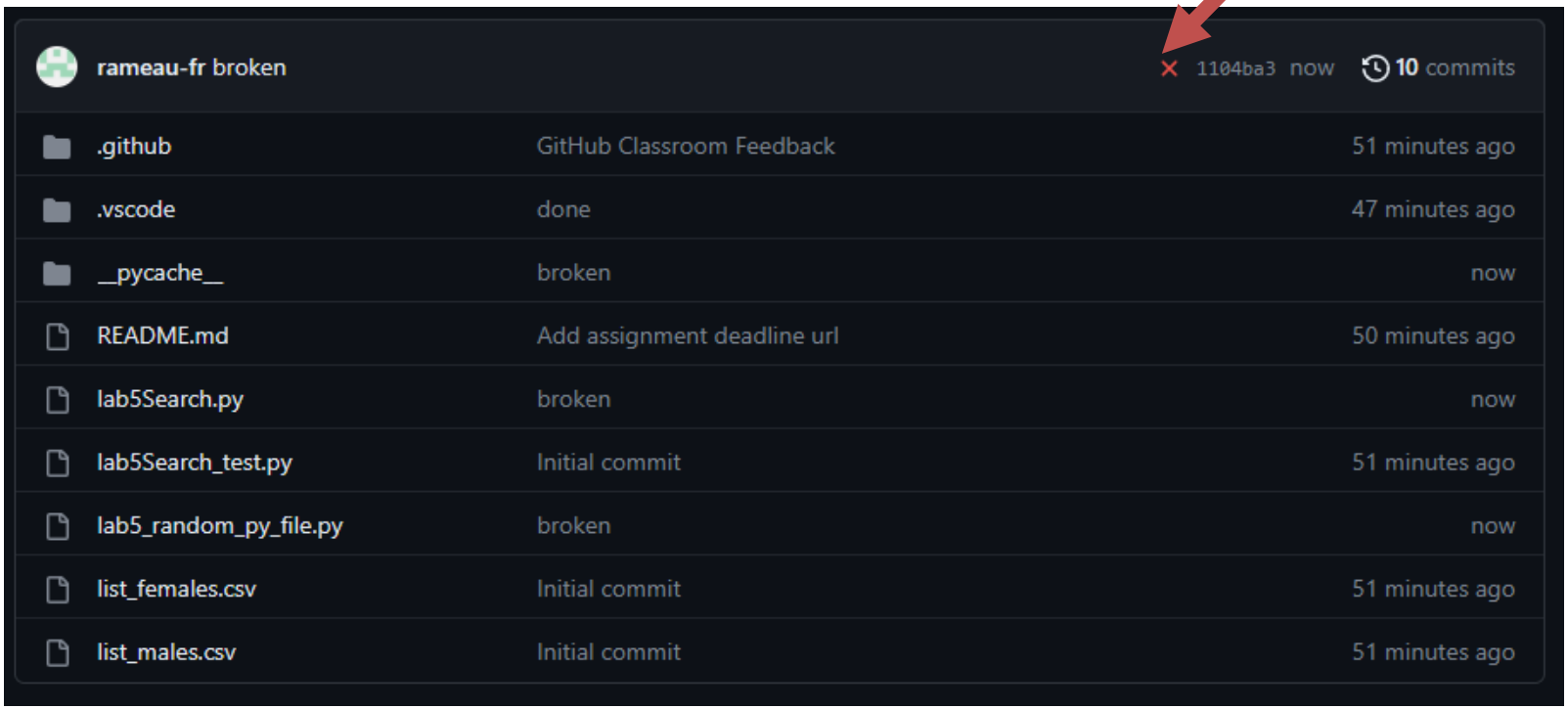
Unit test

Let's push a code on GitHub to see what is happening!!

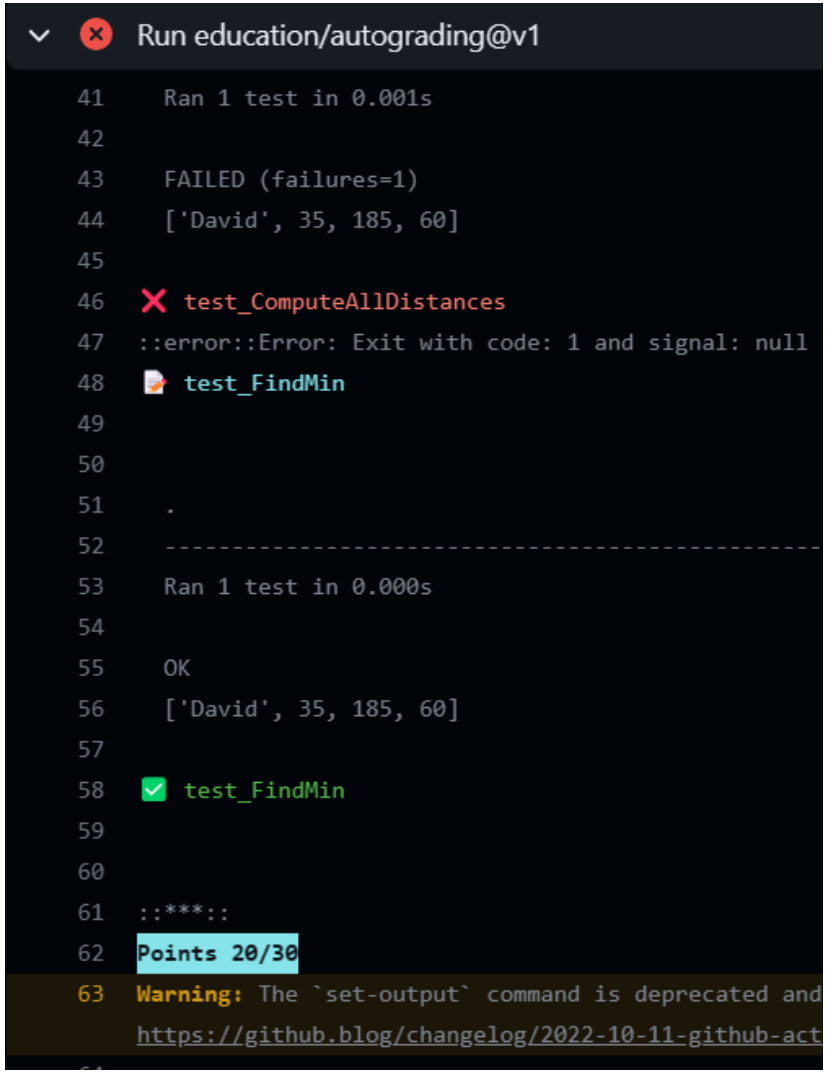
Two possibilities

2.Your code is NOT working as intended

You have this little red cross that means that all your code is NOT working as intended
You can click on it!



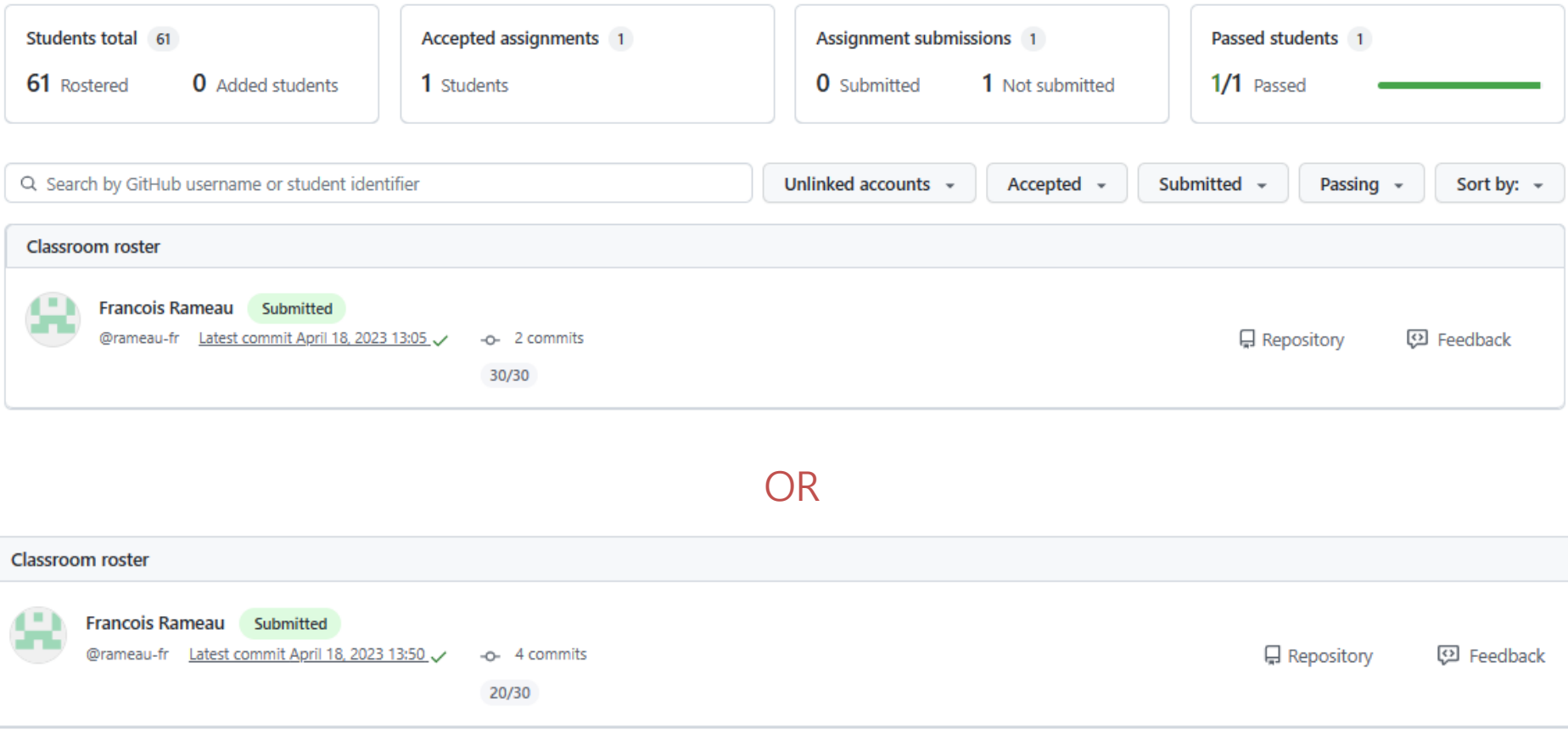
Unit test



You will directly see which tests has been successful or failing AND
Your grade! (of course the TA will ALWAYS double check for assignments)

Unit test

What do I see as an instructor

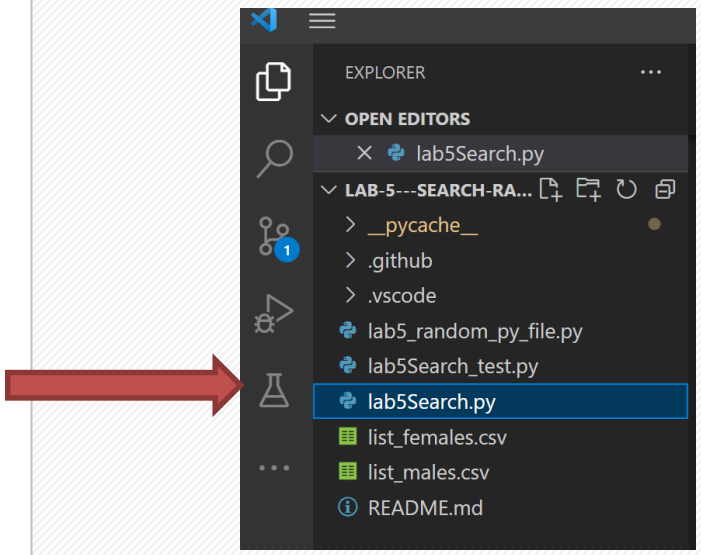


Unit test

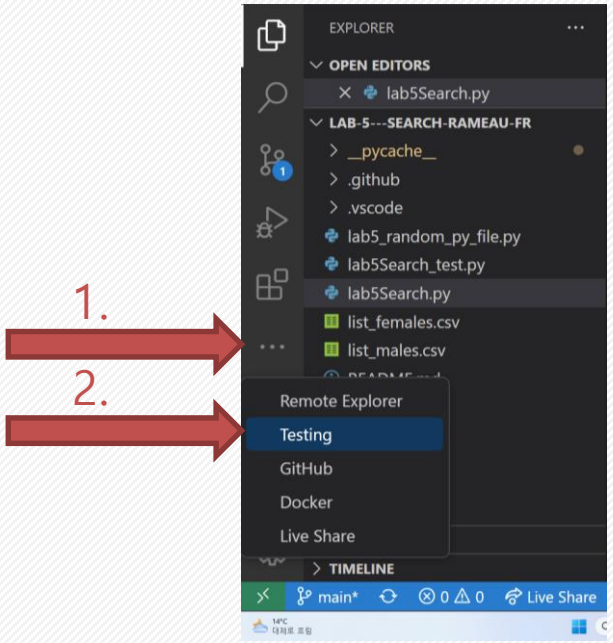
How to run unit test locally

In 5 clicks only!

Step 1: click on Testing



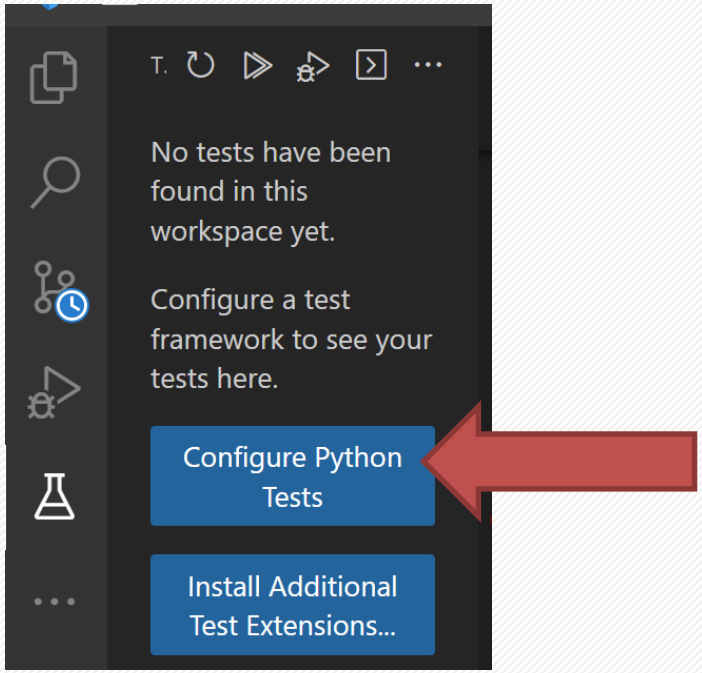
OR



Unit test

How to run unit test locally

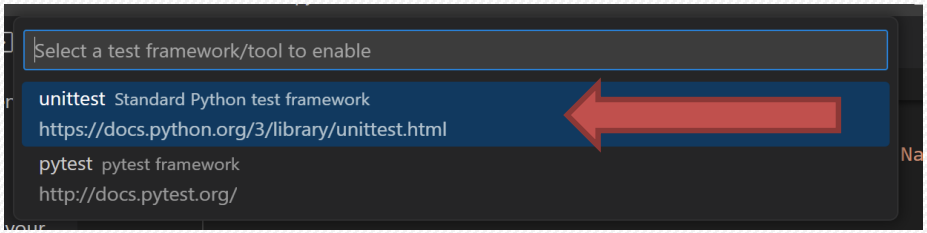
Step 2: configure



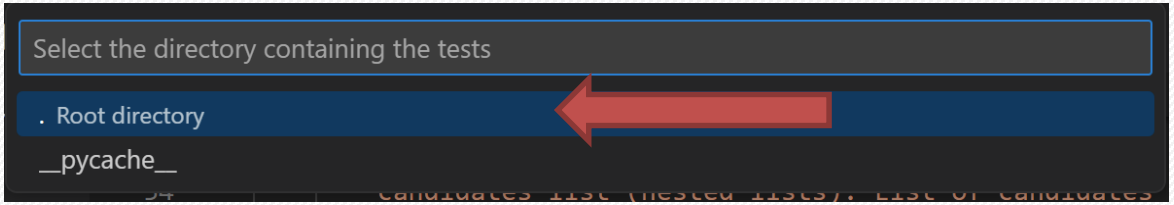
Unit test

How to run unit test locally

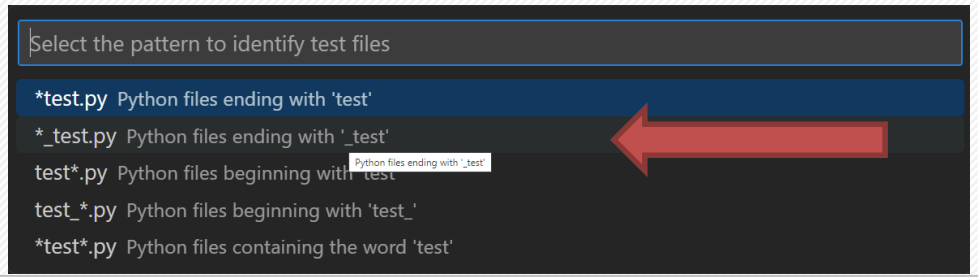
Step 3: Select unittest



Step 4: .Root directory

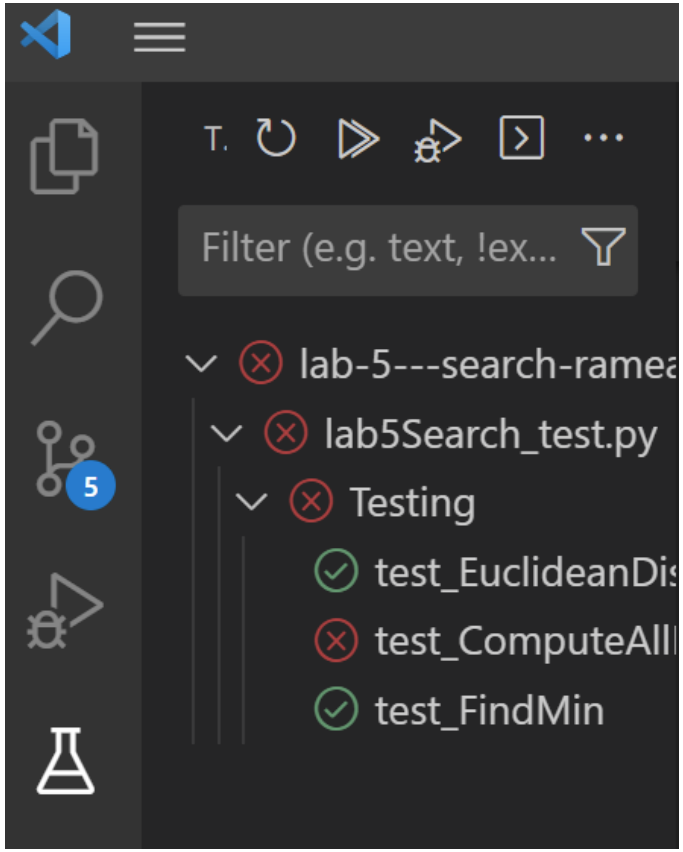


Step 5: *_test.py



Unit test

How to run unit test locally



Now you are ready to go if you want to test your code locally

Feedback via pull request

You now have a branch feedback in your repository, we will use it to send you feedback directly (you will receive an email each time we are trying to reach you out via this mean)