

➤ **Cách 1:** Dùng một phần mềm soạn thảo code bất kì để chạy các **file.py**

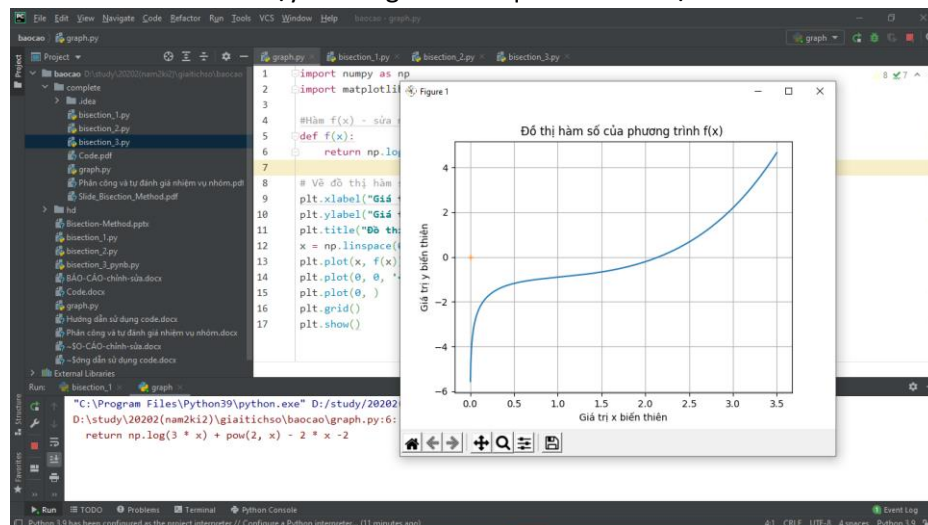
- **Graph.py** : chương trình vẽ đồ thị của một hàm $f(x)$
- **Bisection_1.py** : Tiên nghiệm
- **Bisection_2.py** : Hậu nghiệm
- **Bisection_3.py** : Hậu nghiệm tối ưu

Ở đây mình dùng **PyCharm**

- Đầu tiên bạn mở file: **graph.py** để chạy đồ thị và lấy khoảng cách ly
- Bạn có thể sửa một hàm $f(x)$ bất kì mà bạn muốn ở phần code này

```
graph.py x bisection_1.py x bisection_2.py x bisection_3.py x
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 #Hàm f(x) - sửa một hàm bất kì mà bạn muốn
5 def f(x):
6     return np.log(3 * x) + pow(2, x) - 2 * x - 2
7
8 # Vẽ đồ thị hàm số
9 plt.xlabel("Giá trị x biến thiên")
10 plt.ylabel("Giá trị y biến thiên")
11 plt.title("Đồ thị hàm số của phương trình f(x) ")
12 x = np.linspace(0, 3.5, 1000)
13 plt.plot(x, f(x))
14 plt.plot(0, 0, '+')
15 plt.plot(0, )
16 plt.grid()
17 plt.show()
```

- Sau đó mình sẽ chạy chương trình và quan sát đồ thị:



- Tiếp đó chọn 1 trong 3 file còn lại tùy ý bạn thích để tìm nghiệm hàm $f(x) = 0$
- Ví dụ ở đây mình mở file: **bisection_1.py** (Tiên nghiệm)
(nếu có sửa hàm $f(x)$ ở đồ thị thì các bạn nhớ sửa hàm $f(x)$ ở phần bên này nhé!)
- Chạy chương trình lên và ta bắt đầu nhập (a,b) khoảng cách ly nghiệm, ϵ rồi chờ kết quả thôi!!

Xác định cận dưới a của khoảng cách ly nghiệm.

a = 1

Xác định cận trên b của khoảng cách ly nghiệm.

b = 3

Độ chính xác epsilon.

epsilon = 0.000000001

- Chúng ta có thể quan sát kết quả trả về gồm một bảng hiển thị từng lần lặp, nghiệm x và thời gian chạy chương trình.

The screenshot shows a Python IDE with the following code in `bisection_1.py`:

```

10 return np.log(3 * x) + pow(2, x) - 2 * x - 2
11
12 # Bisection method
13 def bisection():
14     def try_():
15         print("-----")
16         print("Bạn có muốn sử dụng lại chương trình bisection method không? Yes/No? Y/N?")

```

The Run console displays the following table of results:

20	2.1484680176	2.1484699249	2.1484689713	-1.2784e-06	1.6562e-06	1.889e-07
21	2.1484680176	2.1484689713	2.1484684944	-1.2784e-06	1.889e-07	-5.447e-07
22	2.1484684944	2.1484689713	2.1484687328	-5.447e-07	1.889e-07	-1.779e-07
23	2.1484687328	2.1484689713	2.148468852	-1.779e-07	1.889e-07	5.5e-09
24	2.1484687328	2.148468852	2.1484687924	-1.779e-07	5.5e-09	-8.62e-08
25	2.1484687924	2.148468852	2.1484688222	-8.62e-08	5.5e-09	-4.03e-08
26	2.1484688222	2.148468852	2.1484688371	-4.03e-08	5.5e-09	-1.74e-08
27	2.1484688371	2.148468852	2.1484688446	-1.74e-08	5.5e-09	-6e-09
28	2.1484688446	2.148468852	2.1484688483	-6e-09	5.5e-09	-2e-10
29	2.1484688483	2.148468852	2.1484688502	-2e-10	5.5e-09	2.6e-09
30	2.1484688483	2.1484688502	2.1484688492	-2e-10	2.6e-09	1.2e-09
31	2.1484688483	2.1484688492	2.1484688488	-2e-10	1.2e-09	5e-10
32	2.1484688483	2.1484688488	2.1484688486	-2e-10	5e-10	1e-10
33	2.1484688483	2.1484688486	2.1484688484	-2e-10	1e-10	-0.0

Below the table, the console shows:

```

=> Nghiệm gần đúng của phương trình là: x = 2.1484688484
- Time: 0.00029999999640040187 ms
-----
Bạn có muốn sử dụng lại chương trình bisection method không? Yes/No? Y/N?

```

- Bạn có thể tiếp tục hoặc dừng chương trình:
Bạn có muốn sử dụng lại chương trình bisection method không? Yes/No? Y
N
Cảm ơn. Hẹn gặp lại ♥

Process finished with exit code 0

- Với **bisection_2.py**, **bisection_3.py** các bạn làm tương tự.

- **Cách 2:** Truy cập trang: <https://colab.research.google.com/>
(chạy online nên phải có mạng nhé ^^)
- Vào file **code.pdf** rồi copy code hoặc nhấp vào link bên dưới mỗi phần để truy cập vào colab.
 - Ấn **Ctrl+ Enter** để chạy chương trình.

Chúc các bạn học tập vui vẻ !!