Instructions: Name your file hw2.py and submit on CCLE. Add comments to each function.

• Problem 1:

Write a function longestpath(d) that finds the length of the longest path, $(a:b) \rightarrow (b:c) \rightarrow \cdots$, in a dictionary d. It counts each pointer from a key to a value as one step. For example, the path $(a:b) \rightarrow (b:c)$ has length 2. To avoid cycles, we do not allow any key to appear more than once in a path (as a key).

Test cases:

```
d1 = {"a":"b","b":"c"}
d2 = {"a":"b","b":"c","c":"d","e":"a","f":"a","d":"b"}
longestpath(d1) should return 2.
longestpath(d2) should return 5.
```

• Problem 2:

Implement Newton's method (also known as the Newton-Raphson method) to find a root (zero) of a function. No prior knowledge of this algorithm is needed. Just follow the steps.

Given a function f(x), the function's derivative f'(x), and a desired tolerance ϵ (usually a very small positive number), your goal is to find a desired value x^* which is close enough to a root of f(x) such that $|f(x^*)| \leq \epsilon$. The algorithm is as follows:

Algorithm:

- 1. Starting from an initial guess x_0 , calculate the error of your guess $f(x_0)$.
- 2. If $|f(x_0)| \le \epsilon$, then you are done because x_0 is close enough to the root. Otherwise, a better approximation than x_0 is given by $x_1 = x_0 \frac{f(x_0)}{f'(x_0)}$.
- 3. Keep updating your guess x_n using the formula $x_{n+1} = x_n \frac{f(x_n)}{f'(x_n)}$ until you have $|f(x_n)| \le \epsilon$.

Instructions:

– Write your algorithm in a solve function that takes as input a function f(x), its derivative f'(x), an initial guess x_0 and the tolerance ϵ . This function can be called like this:

```
solve(lambda x: [x**2-1, 2*x], 3, 0.0001)
```

- Test your solve function using the following functions f(x), their derivatives f'(x), and initial guesses x_0 :

$$f(x) = x^{2} - 1, f'(x) = 2x, x_{0} = 3$$

$$f(x) = x^{2} - 1, f'(x) = 2x, x_{0} = -1$$

$$f(x) = \exp(x) - 1, f'(x) = \exp(x), x_{0} = 1$$

$$f(x) = \sin(x), f'(x) = \cos(x), x_{0} = 0.5.$$

Use a calculator to test if the solutions provided by your code are correct, and put results in comment in your script.

Suggestions:

- You can start by hard-coding the function, its derivative, the initial guess, and the tolerance in your script without getting user input. This will help you better understand the algorithm.
- If you are still confused, watch this video for an example of Newton's method with two iterations: https://www.youtube.com/watch?v=xdLgTDlFwrc.