

Function-Solving

Introduction

In this, we will use concepts such as searching, sorting, linked lists/dictionary and algorithmic analysis in this assignment.

Assume that you are working for an educational institution, where physics researchers need help to perform some mathematical computations for them so that they can focus on the actual problem and save their time from trivial tasks. The requirement is to develop and provide a code that can do the operations on mathematical functions. Let's assume that there are two functions $f(x)$ and $g(x)$. Your task is to write python programs that can find the addition and multiplication of $f(x)$ and $g(x)$ as an output. Also if a researcher wants to find the middle element of the output equations, they should be able to do it with a minimum number of comparisons. Another requirement is to save the output equations in decreasing order in terms of power.

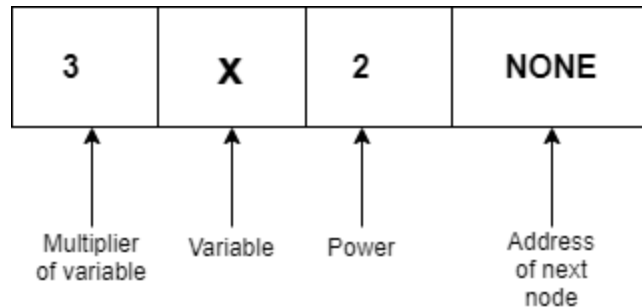
Please find more details related to equation and their output in the **expected_output.txt** file

Housekeeping points

- This is a minimal example and may not follow some standard practices
- The focus is on the main flow, with minimal error handling. Errors in validation logic should be handled appropriately though.
- To keep the scope minimum, please assume that maximum power for input functions will be 2 for $f(x)$ and $g(x)$.
- Also we can assume that both the functions are continuously increasing functions.

Program Organization

- **function_operation.py**: This python file will have the class related to the linked list. It will consist, constructor with some instance variables and class variables if required. Creating an object for this class means, you have successfully created the nodes that will have information of each part of the equation.
 - **`__init__()`**: Define the constructor for the class with appropriate instance variables. The idea is that each node will have four information as mentioned in the following diagram:



This above diagram can be treated as $3 \cdot x^2$ in function format.

Where * means multiplication and ^ is the notation for power. This method is already implemented.

- **main.py:** This is the main driver program. It creates objects that will hold the initial heads for $f(x)$ and $g(x)$. This python will consist of all the operations related to those equations such as addition, multiplication, finding the middle element, sorting the output list etc.
- **models.py:** In this python file all the implementations related to the task are mentioned. You should implement your logic in this file. There are methods named as addition, multiplication, addition and sort are available. Internally these methods will call LinkedList methods available in function_operation.py.

Problem Statement

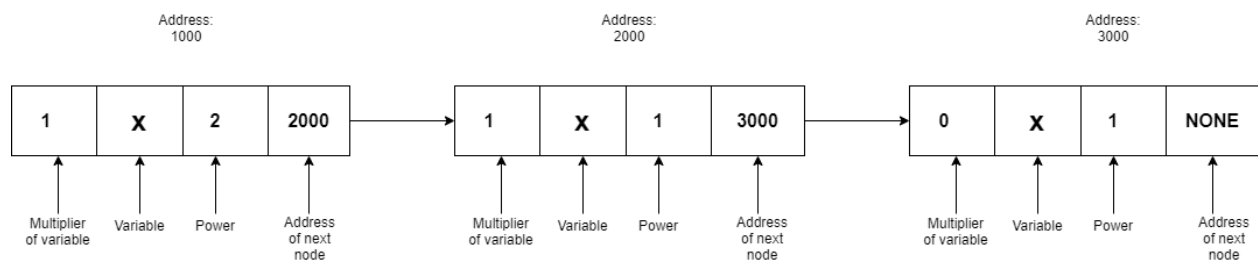
The program structure is already set and there are specific methods that you are expected to implement. Please also read the comments in the code, especially in the methods to be implemented. You can modify main.py to add further demonstration calls if you want or add another caller file to do further tests if preferred.

1. Implement the models python file:
 - a. **addition():** This method will take the heads of the two equations and perform the addition operation for you. Please look at the **expected_output.txt** to get a clear understanding.
 - i. Take care of the fact that nodes that have the same power can only be added and if there is a node that is available only in one of the equations then the output equation will have that node automatically.
 - b. **multiplication():** Similar to the addition method, this will also take heads of two equations as a parameter and it will perform the multiplication on those two equations.
 - i. After doing the multiplication it is possible to have the nodes with different coeff and same power of x. You have to make sure that if the power of

two nodes are the same then their coeff will be added and extra nodes will be deleted or dropped from the linked list.

- c. **sorting_function():** This method will take the equation head as a parameter and sort the equation based on power of x in decreasing order. Please have a look at the output file to get more context.
2. In the function_operation.py python file, you are supposed to implement a method inside the class that will perform the sorting operation on the given linked list based on the power of x. More details related to this are available in the attached output file.
 - a. **sort_linked_list:** Implement sorting function in the class to perform sorting. Please do not worry about the time complexity for this algorithm.

Function example:



Now $f(x)$ can be treated or written as : $1 \cdot x^2 + 1 \cdot x^1 + 1$

Evaluation Rubric

Total Project Points: **100**

- Basic compilation without errors (10%) : **10 Points**
- Correctness:
 - Problem statement - 1.a (25%) : **25 Points**
 - Problem statement - 1.b (25%) : **25 Points**
 - Problem statement - 1.c (20%) : **20 Points**
 - Problem statement - 2.a (20%) : **20 Points**

Program Instructions

1. Download the zipped folder named **M01-P02-DSA-Function-Solving.zip**, and unzip it on your local machine. Go into the directory named **M01-P02-DSA-Function-Solving**.
2. Make sure that you have Python 3.6, or higher, installed. At your command prompt, run:

```
$ python --version  
Python 3.7.3
```

If not installed, install the latest available version of Python 3.

3. To run the code in the source code folder, run the following command:

```
$ python3 main.py (On many Linux/Mac platforms)  
OR  
$ python main.py (On Windows/Mac platforms)
```

In any case, one of these two commands should work.

4. Alternatively, you could install a popular Python IDE, such as PyCharm or Visual Studio Code, and select a command to build the project from there.
5. You will be making very frequent changes into the **main.py** and **models.py** and **function_operation.py** python files. The idea is to complete both the scripts so that it satisfies all the requirements. Once you have completed the changes in the code and it is executed without any error. Zip the folder as **M01-P02-DSA-Function-Solving_Solution.zip** & now it is ready for submission.

References

- a. <https://www.mathsisfun.com/sets/functions-operations.html>
- b. <https://www.purplemath.com/modules/fcnops.htm>