

## RE: EN.605.202.81.SP21, Lab 3: Considering using a dictionary to dynamically allocate variables to node

Eleanor Chlan <echlan1@jhu.edu>

Thu 2021-04-08 3:53 PM

To: Peter Rasmussen <prasmus3@jhu.edu>

The two Scotts and I have discussed it and we are okay with you using a Python dictionary for the node only.

Thanks,

*Eleanor*

*Lead Adviser AI-CS-CyS-DS-ISE*

If you are looking for advising help, please email [advisingCS-CyS-ISE@jhu.edu](mailto:advisingCS-CyS-ISE@jhu.edu)

Please email me to schedule a phone call. I am generally available for calls 11-4pm on weekdays.

If you are a student in one of my courses, please use [echlan1@jhu.edu](mailto:echlan1@jhu.edu)

---

**From:** Peter Rasmussen <prasmus3@jhu.edu>

**Sent:** Thursday, April 8, 2021 11:56 AM

**To:** Eleanor Chlan <echlan1@jhu.edu>

**Subject:** Re: EN.605.202.81.SP21, Lab 3: Considering using a dictionary to dynamically allocate variables to node

That's a good point.

I wonder if I could do the following to address your point.

- I have two terms:
  - $A = xy$
  - $B = 2xy + x^2$
- I want to add the terms  $A + B = \text{answer}$ .
- I create a circular linked list of three nodes: A, B, and answer
- Each node is structured as follows:
  - Node A
    - dictionary  $dA = \{ 'xy': 1 \}$  // making sure terms are ordered alphabetically while preserving exponent order
    - pointer next = Node B
    - string next\_op = '+'
  - Node B
    - dictionary  $dB = \{ 'xy': 2, 'x^2': 1 \}$
    - pointer next = Node answer
    - string next\_op = '='
  - Node answer
    - dictionary  $danswer = \{ \}$  // initially
    - dictionary  $danswer = \{ 'xy': 3, 'x^2': 1 \}$  // ultimately

- I iterate over the string A + B and build and populate the nodes accordingly. If I do it this way I could evaluate cases where there are more than two term nodes (say I have A + B + C).
- After populating the nodes I cycle through the circular list to build the answer node
- I echo the input and print the output symbols and the evaluated answer
  - $A + B = xy + 2xy + x^2 = 3xy + x^2 = \text{whatever answer gets evaluated using input values for } x \text{ and } y$

Best,

Peter

Peter Rasmussen  
Data Scientist  
202.644.1575

---

**From:** Eleanor Chlan <[echlan1@jhu.edu](mailto:echlan1@jhu.edu)>  
**Sent:** Thursday, April 8, 2021 10:23 AM  
**To:** Peter Rasmussen <[prasmus3@jhu.edu](mailto:prasmus3@jhu.edu)>  
**Subject:** RE: EN.605.202.81.SP21, Lab 3: Considering using a dictionary to dynamically allocate variables to node

In the general scheme of thing, I like the way you are thinking, but I am concerned about matching the exponents: x to x and y to y. Would you still have fixed positions or would ewach node be different. I will get back to you on this.

Thanks,

*Eleanor*

*Lead Adviser AI-CS-CyS-DS-ISE*

If you are looking for advising help , please email [advisingCS-CyS-ISE@jhu.edu](mailto:advisingCS-CyS-ISE@jhu.edu)

Please email me to schedule a phone call. I am generally available for calls 11-4pm on weekdays.

If you are a student in one of my courses, please use [echlan1@jhu.edu](mailto:echlan1@jhu.edu)

---

**From:** Peter Rasmussen <[prasmus3@jhu.edu](mailto:prasmus3@jhu.edu)>  
**Sent:** Thursday, April 8, 2021 7:11 AM  
**To:** Eleanor Chlan <[echlan1@jhu.edu](mailto:echlan1@jhu.edu)>  
**Subject:** EN.605.202.81.SP21, Lab 3: Considering using a dictionary to dynamically allocate variables to node

Hi Eleanor,

I'm thinking of using a dictionary to dynamically allocate variables to a node, which would allow the user to specify any number of variables (coefficient, a, b, c, x, y, z) rather than just four (coefficient, x, y, z).

I could use Python's in-built dictionary, but I'm not sure if that's allowed. You can instantiate an empty dictionary in Python by simply assigning a variable, say d, to `d = {}`. If we're not allowed to use the built-in dictionary, I could build my own string hash function using something found on the internet (e.g., [this one](#)).

If we're not allowed to use dictionaries, I could always just bind node terms to the object (e.g., `node.x = 1`, and then account for case when `node.x=1` for one node and `node.x!=1` in another) but I think a dictionary may be more elegant.

Do you think a dictionary is a reasonable choice to hold variables in a node?

Best,

Peter Rasmussen  
Data Scientist  
202.644.1575

**From:** Peter Rasmussen  
**Sent:** Thursday, April 8, 2021 6:00 AM  
**To:** Eleanor Chlan <[echlan1@jhu.edu](mailto:echlan1@jhu.edu)>  
**Subject:** EN.605.202.81.SP21, Lab 3: Method of evaluation of terms

Hi Eleanor,

After we consolidate polynomials we need to evaluate them.

- Let's say we have  $A=xy + x$  and  $B=2xy$ ,  $x=1$  and  $y=2$ , and we want to add  $A + B$ .
- We consolidate terms and get  $A + B = 3xy + x$
- Can we then use regular multiplication to evaluate the first term  $3xy=3*1*2$ ?
- Or do we need to use a more primitive method whereby we take the sum of sums:

$$3xy = \sum_{i=1}^3 \sum_{j=1}^{y=2} x = \sum_{i=1}^3 \sum_{j=1}^{y=2} 1 = 1 + 1 + 1 + 1 + 1 + 1 = 6$$

Best,

Peter

Peter Rasmussen  
Data Scientist  
202.644.1575