**Introduction:**

My job was to design and implement a data structure called [Least Recently Used (LRU)](https://en.wikipedia.org/wiki/Cache_replacement_policies#LRU) cache. This data structure supports the following operations

get(key) - Gets the value (will always be positive) of the key if the key exists in the cache, otherwise return -1.

put(key, value) - Insert or replace the value if the given key is not already in the cache. When the cache reaches its maximum capacity, it should invalidate the **least recently used item** before inserting a new item.

size() – Returns the number of key/value pairs currently stored in the cache

max\_capacity() – Returns the maximum capacity of the cache

All operations MUST run in O(1) time complexity. You are free to uses Python’s set and/or dictionary data structures. If you need to use a doubly linked list (hint), you need to code it yourself.

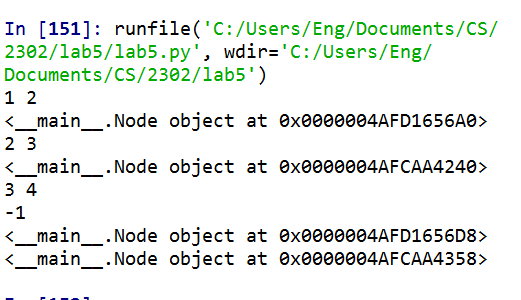
**Proposed solution design and implementation:**

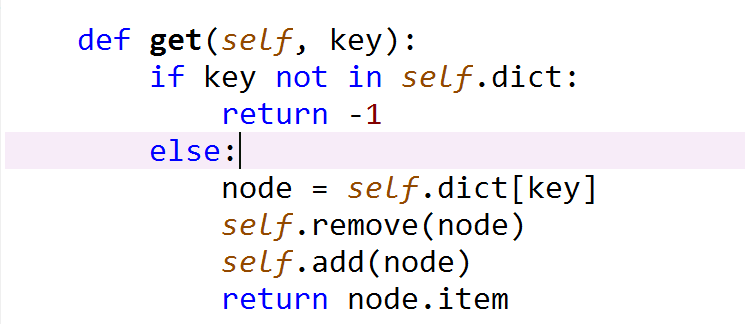
In order of finishing this method first of all I got to study and see how LRU cache works, at the same time I got to implement all the methods they told me before, after doing that I asked my professor in order of knowing how can I be using doubly linked list and hash map at the same time.

For doing my lab I created a node class and a ULR cache(containg all the information in the hashmap key).

I got to sorted the nodes using a doubly linked list, the hashmap was used in order of replacing the last used to the newest one. The comparation between two nodes where mainly by its key. I used 2 more methods as helpers, add and remove method, a printing method and two helper pointers in order of tracing faster and easier head and tails.

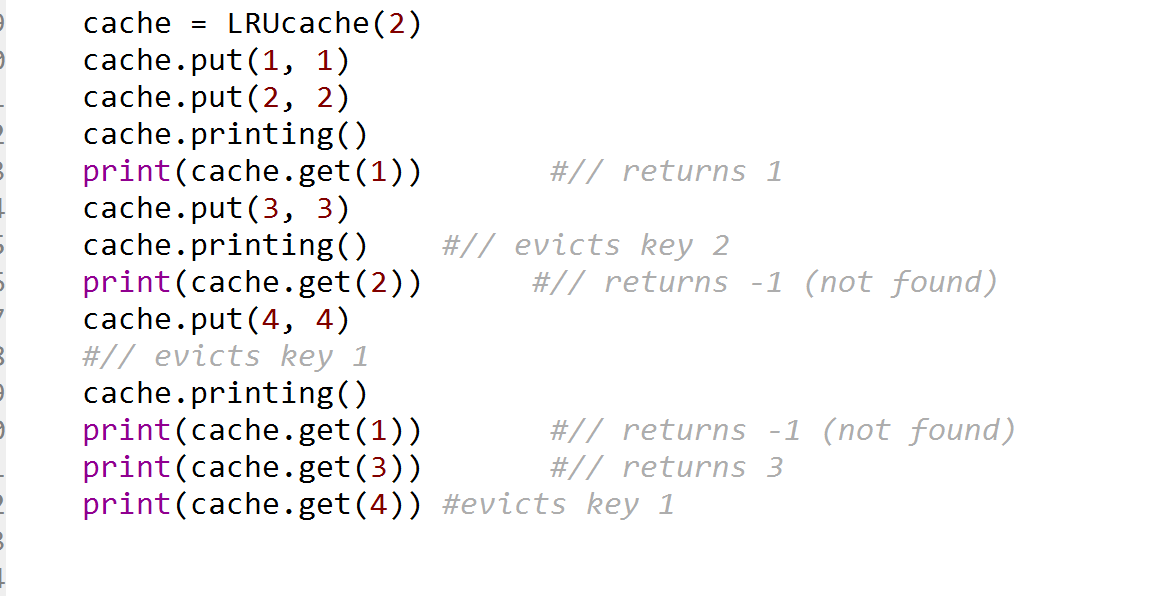
**Experimental results:**

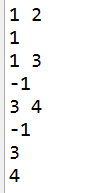
I received a problem at first because I was not able to finish this lab because of the transversing of the get method

Then one of my upper level classmates help me in order to know how the get method worked, and I came with this.

After receiving help, I finished my lab, however I got stocked in the same comment so I sent an e-mail to my T.A. in order of helping me out.

Source Method:



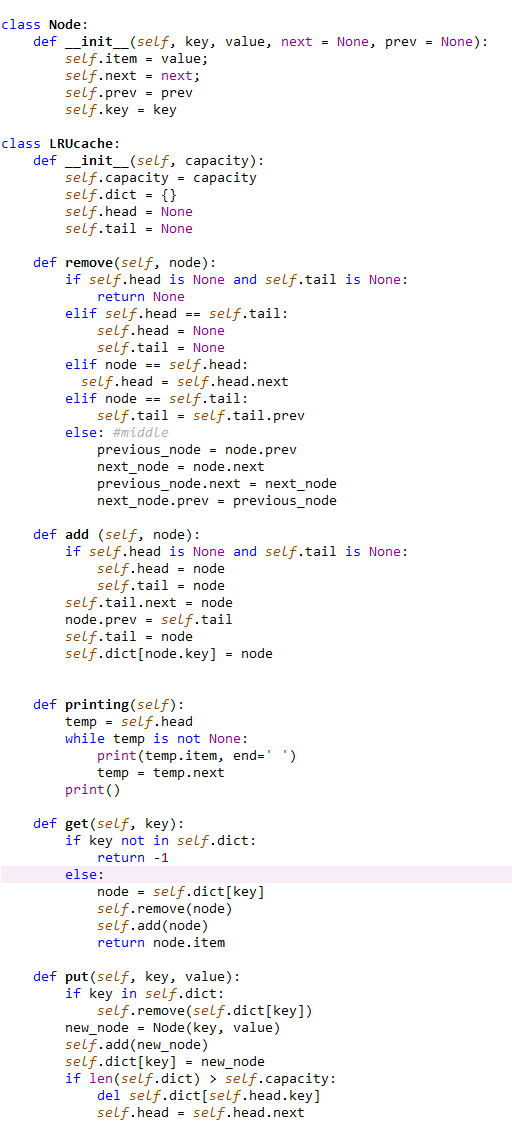


Finally, I went to my solution because after traversing into the doubly linke list in the get method it takes the node send it and move it into the first node, instead of just taking it and print it.

**Conclusion:**

I felt confident at the time finishing my lab because I never taught that I could transverse into a doubly linked list; this is the first time I used this two data structures together,

**Source Code:** method



**Academic Honesty**

“I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”