

OPEN ENDED LAB
CS-218 DATA STRUCTURES & ALGORITHMS
SE CS BATCH 2023
FALL SEMESTER 2024

PROBLEM DESCRIPTION:

Design a data structure in Python that follows the constraints of a *Least Recently Used (LRU)* cache and find its time and space complexities.

Implement the LRUCache class:

LRUCache(int capacity) Initialize the LRU cache with positive size capacity.

int get(int key) Return the value of the key if the key exists, otherwise return -1.

void put(int key, int value) Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, evict the least recently used key. Each call to put and get functions is counted a reference.

Example:

Input

```
["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"]  
[[2], [1, 1], [2, 2], [1], [3, 3], [2], [4, 4], [1], [3], [4]]
```

Output

```
[null, null, null, 1, null, -1, null, -1, 3, 4]
```

Explanation

```
LRUCache lRUCache = new LRUCache(2);  
lRUCache.put(1, 1); // cache is {1=1}  
lRUCache.put(2, 2); // cache is {1=1, 2=2}  
lRUCache.get(1);    // return 1  
lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}  
lRUCache.get(2);    // returns -1 (not found)  
lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}  
lRUCache.get(1);    // return -1 (not found)  
lRUCache.get(3);    // return 3  
lRUCache.get(4);    // return 4
```

Constraints:

1 <= capacity <= 50

0 <= key <= 100

0 <= value <= 100

Test the above task by filling the full cache using keys 0-49. Retrieve the odd number key values. Fill the cache with prime number keys 0-100. In the end, compute the final miss rate. Miss rate needs to be calculated in the program. Your program should be able to keep track of the misses happened at any point and should return the miss rate when you call for it.

The task will be graded for CLO-3: **Practice** with algorithms for widely used computing operations, C3, PLO-3 using the rubric at the end of this file.

INSTRUCTIONS

1. Students can make groups of **at most three** students for this assignment.
2. Groups can only be made **within a practical group**.
3. A project is being offered by the teacher (see Google Classroom). You can add any additional features to the project to make it distinct and stand out.
4. Submit your file(s) in a zip folder with the following naming convention: GX-YY_CEP, where X represents your practical group and YY represent your group number. For example: G3-12_CEP.zip
5. The folder will contain
 - Main code file(s)
 - Report
 - ReadMe (optional)
6. In case of non-compliance, the assignment will not be graded.
7. Your assignment will be graded on the attached rubrics below

DELIVERABLES

1. Submission deadline is 25th Nov 2024.
2. Project will be evaluated in Week 15 during your lab hours.
3. Bring your laptop and report in hardcopy for project evaluation.
4. Content to be covered in report (organize in headings):
 - a. Problem Description
 - b. Flow of your project which includes highlighting the data structure and/or algorithm used to solve the problem
 - c. Most challenging part for you while working on the project
 - d. Any new thing learnt in Python while working on the project
 - e. At least 3 test case runs. Screen shots should be attached.
5. While compiling the report, keep the following in mind:
 - a. Attach the provided rubric on top of your report, with names and roll numbers filled in.
 - b. No need for any other title page, the rubric sheet will serve as the title page.
 - c. Use font size of 12 and/or 14 for headings, and 11 for regular text.
 - d. Font style should be Times New Roman.

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

Course Code: CS-218

Course Title: Data Structures & Algorithms

Complex Engineering Problem

SE Batch 2023, Fall Semester 2024

Grading Rubric

TERM PROJECT Group

Members:

Student No.	Name	Roll No.
S1		
S2		
S3		

CRITERIA AND SCALES				Marks Obtained		
				S1	S2	S3
Criterion 1: Has the student provided the appropriate design of LRU data structure?						
0	1	2	-			
The chosen design is too simple	The design is fit to be chosen for a class project	The choice is different and impressive.	-			
Criterion 2: How good is the programming implementation?						
0	1	2	3			
The project could not be implemented	The project has been implemented partially.	The project has been implemented completely but can be improved.	The project has been implemented completely and impressively			
Criterion 3: How well written is the report?						
0	1	2	-			
The submitted report is unfit to be graded	The report is partially acceptable	The report is complete and concise				
Total Marks:						