

1. 파이썬의 라이브러리인 배열을 이용해 LRU Simulator 코드를 완성하시오.
템플릿 코드: https://github.com/eunjicious/ds_2024.git

소스코드

```
class CacheSimulator:
    def __init__(self, cache_slots):
        self.cache_slots = cache_slots
        self.cache_hit = 0
        self.tot_cnt = 0
        self.cache = []

    def do_sim(self, page):
        self.tot_cnt += 1
        if page in self.cache:
            # 페이지가 캐시에 이미 있으면, 캐시 히트이며, 해당 페이지를 캐시의 끝으로
            이동 -> python 내부에서 append 를 어떻게 최적화 할지는 모르지만 만약 [] 가 C++ 기준
            배열이라면, 비효율적인 동작, 만약 연결 리스트 방식이면, 그나마 효율이 있다
            self.cache_hit += 1
            self.cache.append(self.cache.pop(self.cache.index(page)))
        else:
            # 페이지가 캐시에 없으면, 새로운 페이지를 추가
            if len(self.cache) >= self.cache_slots:
                # 캐시가 가득 차 있으면, 가장 오래된 페이지(리스트의 첫 번째 요소)를
                제거 -> 이 역시 동일하게 효율이 아쉽다.(내부 구현을 모르기에 어떻게 작동하는지를
                모르겠음)
                self.cache.pop(0)
            self.cache.append(page)

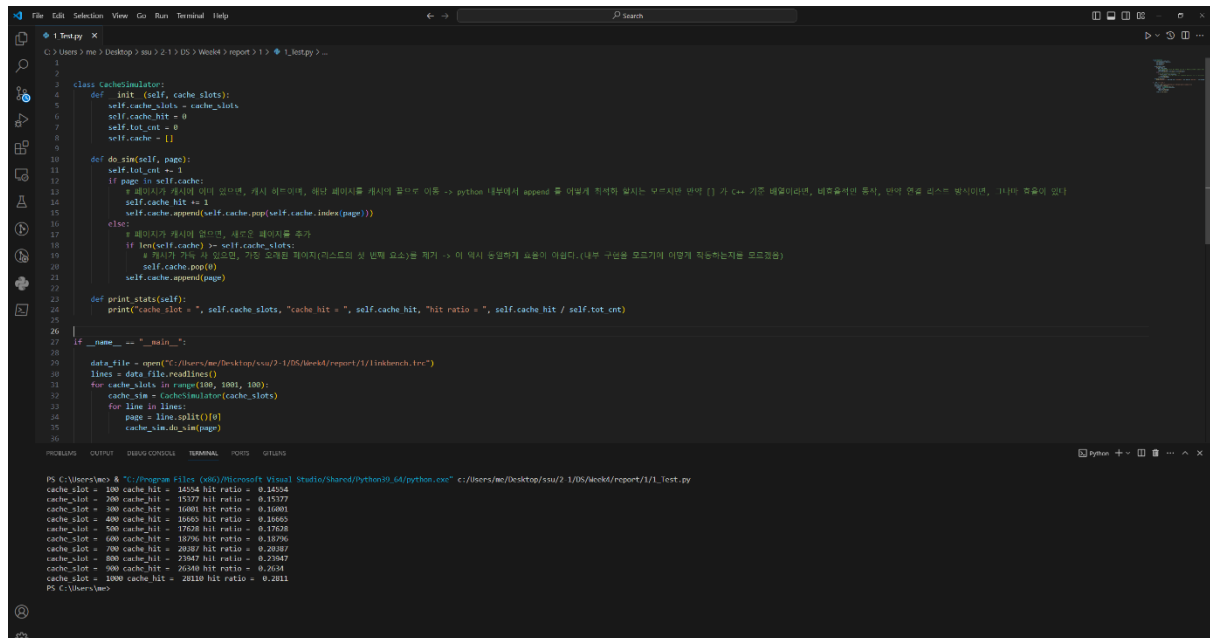
    def print_stats(self):
        print("cache_slot = ", self.cache_slots, "cache_hit = ",
              self.cache_hit, "hit ratio = ", self.cache_hit / self.tot_cnt)

if __name__ == "__main__":

    data_file = open("C:/Users/me/Desktop/ssu/2-
1/DS/Week4/report/1/linkbench.trc")
    lines = data_file.readlines()
    for cache_slots in range(100, 1001, 100):
        cache_sim = CacheSimulator(cache_slots)
        for line in lines:
            page = line.split()[0]
            cache_sim.do_sim(page)
```

```
cache_sim.print_stats()
```

결과물



The screenshot shows a Python IDE with a file named `cache_sim.py`. The code defines a `CacheSimulator` class with methods `__init__`, `do_sim`, and `print_stats`. The `do_sim` method simulates a cache with a given number of slots, processing a list of pages. The `print_stats` method prints the cache slot size, cache hits, and hit ratio. The main function reads a file `l1linbench.txt` and runs the simulator for cache slot sizes from 100 to 1000.

```
1  
2  
3 class CacheSimulator:  
4     def __init__(self, cache_slots):  
5         self.cache_slots = cache_slots  
6         self.cache_hit = 0  
7         self.tot_cnt = 0  
8         self.cache = []  
9  
10    def do_sim(self, page):  
11        self.tot_cnt += 1  
12        if page in self.cache:  
13            # 페이지가 캐시에 이미 있으면, 캐시 히트이며, 해당 페이지를 캐시에 곁으로 이동 -> python 내부에서 append 후 어떻게 처리와 없지는 모르지만 만약 [] 가 ++ 가중 배열이라면, 배열출력만 출력, 만약 연속 리스트가 발생하면, 1차원 배열이 있다  
14            self.cache_hit += 1  
15            self.cache.append(self.cache.pop(self.cache.index(page)))  
16        else:  
17            # 페이지가 캐시에 없으면, 새로운 페이지를 추가  
18            if len(self.cache) >= self.cache_slots:  
19                # 캐시가 가득 찼고 있으면, 가장 오래된 페이지(가장 오래된 것 번째 요소)를 제거 -> 이 역시 동일하게 구현이 가능하다. (내부 구현은 요요기에 어떻게 작동하는지를 모르겠음)  
20                self.cache.pop(0)  
21            self.cache.append(page)  
22  
23    def print_stats(self):  
24        print("cache slot = ", self.cache_slots, "cache hit = ", self.cache_hit, "hit ratio = ", self.cache_hit / self.tot_cnt)  
25  
26  
27 if __name__ == "__main__":  
28     data_file = open("C:/Users/me/Desktop/ssu/2-1/D5/Week4/report/1/l1linbench.txt")  
29     lines = data_file.readlines()  
30     for cache_slots in range(100, 1001, 100):  
31         cache_sim = CacheSimulator(cache_slots)  
32         for line in lines:  
33             page = line.split()[0]  
34             cache_sim.do_sim(page)  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

The terminal output shows the results of the simulation for various cache slot sizes:

```
PS C:\Users\me> "C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python9_64/python.exe" c:/Users/me/Desktop/ssu/2-1/D5/Week4/report/1/l1linbench.txt  
cache_slot = 100 cache_hit = 14554 hit ratio = 0.14554  
cache_slot = 200 cache_hit = 15377 hit ratio = 0.15377  
cache_slot = 300 cache_hit = 16001 hit ratio = 0.16001  
cache_slot = 400 cache_hit = 16665 hit ratio = 0.16665  
cache_slot = 500 cache_hit = 17628 hit ratio = 0.17628  
cache_slot = 600 cache_hit = 18796 hit ratio = 0.18796  
cache_slot = 700 cache_hit = 20387 hit ratio = 0.20387  
cache_slot = 800 cache_hit = 23947 hit ratio = 0.23947  
cache_slot = 900 cache_hit = 26340 hit ratio = 0.2634  
cache_slot = 1000 cache_hit = 28110 hit ratio = 0.2811  
PS C:\Users\me>
```

Git push 화면

sapphireforce / DS	
Code Issues Pull requests Actions Projects Security Insights Settings	
main DS / Week4 / report / 1/	
sapphireforce Init	
Name	Last commit message
..	
1.PNG	Init
1_Test.py	Init
1_상세.PNG	Init
linkbench.trc	Init

간단한 설명

가장 단순한 방식으로 밀어넣고 삭제하고 옮기기로 구현했습니다. 속도가 빠를 것 같지는 않을 것 같습니다.

- 파이썬의 연결리스트 (**LinkedListBasic** 에 **tail** 을 추가한 버전 또는 **CircularLinkedList**) 을 이용해 LRU Simulator 코드를 완성하시오.

소스코드

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None

class CircularLinkedList:
    def __init__(self):
        self.tail = None
        self.index = {}

    def append(self, value):
        if value in self.index:
            self.move_to_end(value) # 끝으로 이동
        else:
            NewNode = Node(value)
            if not self.tail: # 없으면 새로 만들어주기
                self.tail = NewNode
                self.tail.next = NewNode
            else: # 아니면 꼬리 물어주기
                NewNode.next = self.tail.next
                self.tail.next = NewNode
                self.tail = NewNode
```

```

        self.index[value] = NewNode

def remove_Oldest(self):
    if not self.tail:
        return None
    Oldest = self.tail.next
    if Oldest == self.tail:
        self.tail = None
    else:
        self.tail.next = Oldest.next
    del self.index[Oldest.value] #지우기(중요 -> 버그 발생 지역 예상)
    return Oldest.value

def move_to_end(self, value): #버그 다발 지역(ㅎㅎㅎ)
    if value in self.index and self.index[value] != self.tail:
        node = self.index[value]
        if node.next == self.tail.next:
            self.tail = node
        else:
            prev_node = self._get_prev_node(node)
            prev_node.next = node.next
            node.next = self.tail.next
            self.tail.next = node
            self.tail = node

#-----#
def _get_prev_node(self, node):
    current = self.tail
    while current and current.next != node:
        current = current.next
    return current

#-----#

class CacheSimulator:
    def __init__(self, cache_slots):
        self.cache_slots = cache_slots
        self.cache_hit = 0
        self.tot_cnt = 0
        self.cache = CircularLinkedList() #왜 안되는겨? -> 한참을 찾았네

    def do_sim(self, page):
        self.tot_cnt += 1
        if page in self.cache.index:
            self.cache_hit += 1
        else:
            if len(self.cache.index) >= self.cache_slots:
                self.cache.remove_Oldest()
            self.cache.append(page)

```

```

def print_stats(self):
    hit_ratio = self.cache_hit / self.tot_cnt if self.tot_cnt > 0 else 0
    print(f"cache_slot = {self.cache_slots}, cache_hit = {self.cache_hit},
hit ratio = {hit_ratio:.4f}")

if __name__ == "__main__":

    data_file = open("C:/Users/me/Desktop/ssu/2-
1/DS/Week4/report/2/lru_sim/linkbench.trc")
    lines = data_file.readlines()
    for cache_slots in range(100, 1001, 100):
        cache_sim = CacheSimulator(cache_slots)
        for line in lines:
            page = line.split()[0]
            cache_sim.do_sim(page)

        cache_sim.print_stats()
#근데 hit rate 가 이정도 나오는게 맞나? 모르겠네

```

결과물

```

class CircularLinkedList:
    def __init__(self):
        self.current = None

    def append(self, data):
        new_node = Node(data)
        if self.current is None:
            self.current = new_node
        else:
            new_node.next = self.current
            self.current.next = new_node
            self.current = new_node

    def __str__(self):
        nodes = []
        current = self.current
        while current:
            nodes.append(current.data)
            current = current.next
        return str(nodes)

class CacheSimulator:
    def __init__(self, cache_slots):
        self.cache_slots = cache_slots
        self.cache_hit = 0
        self.tot_cnt = 0
        self.cache = CircularLinkedList()

    def do_sim(self, page):
        self.tot_cnt += 1
        if page in self.cache.index:
            self.cache_hit += 1
        else:
            if len(self.cache.index) >= self.cache_slots:
                self.cache.remove_oldest()
            self.cache.append(page)

    def print_stats(self):
        hit_ratio = self.cache_hit / self.tot_cnt if self.tot_cnt > 0 else 0
        print(f"cache_slot = {self.cache_slots}, cache_hit = {self.cache_hit}, hit ratio = {hit_ratio:.4f}")

if __name__ == "__main__":
    data_file = open("C:/Users/me/Desktop/ssu/2-1/DS/Week4/report/2/linkbench.trc")
    lines = data_file.readlines()
    for cache_slots in range(100, 1001, 100):
        cache_sim = CacheSimulator(cache_slots)
        for line in lines:
            page = line.split()[0]
            cache_sim.do_sim(page)

        cache_sim.print_stats()

```

```

PS C:\Users\me> python .\lru_sim.py
cache_slot = 100, cache_hit = 1654, hit ratio = 0.1655
cache_slot = 200, cache_hit = 1537, hit ratio = 0.1538
cache_slot = 300, cache_hit = 1601, hit ratio = 0.1601
cache_slot = 400, cache_hit = 1665, hit ratio = 0.1666
cache_slot = 500, cache_hit = 1728, hit ratio = 0.1763
cache_slot = 600, cache_hit = 1806, hit ratio = 0.1808
cache_slot = 700, cache_hit = 2038, hit ratio = 0.2039
cache_slot = 800, cache_hit = 2362, hit ratio = 0.2395
cache_slot = 900, cache_hit = 2636, hit ratio = 0.2634
cache_slot = 1000, cache_hit = 2810, hit ratio = 0.2811
PS C:\Users\me>

```

```
PS C:\Users\me> & "C:/Program Files (x86)/Microsoft Visual Studio
cache_slot = 100, cache_hit = 14554, hit ratio = 0.1455
cache_slot = 200, cache_hit = 15377, hit ratio = 0.1538
cache_slot = 300, cache_hit = 16001, hit ratio = 0.1600
cache_slot = 400, cache_hit = 16665, hit ratio = 0.1666
cache_slot = 500, cache_hit = 17628, hit ratio = 0.1763
cache_slot = 600, cache_hit = 18796, hit ratio = 0.1880
cache_slot = 700, cache_hit = 20387, hit ratio = 0.2039
cache_slot = 800, cache_hit = 23947, hit ratio = 0.2395
cache_slot = 900, cache_hit = 26340, hit ratio = 0.2634
cache_slot = 1000, cache_hit = 28110, hit ratio = 0.2811
PS C:\Users\me>
```

Git push 화면

sapphireforce Init	
Name	Last commit message
..	
lru_sim	Init
2.PNG	Init
2.py	Init
linkbench.trc	Init

간단한 설명

전에 작성한 코드를 최대한 참조해서 만들어 봤습니다.

뭔가 빼먹은 버그나 이상한 작동이 분명히 있을 것 같은데 안보여서 조금은 걱정됩니다.