

## 프로젝트 1.

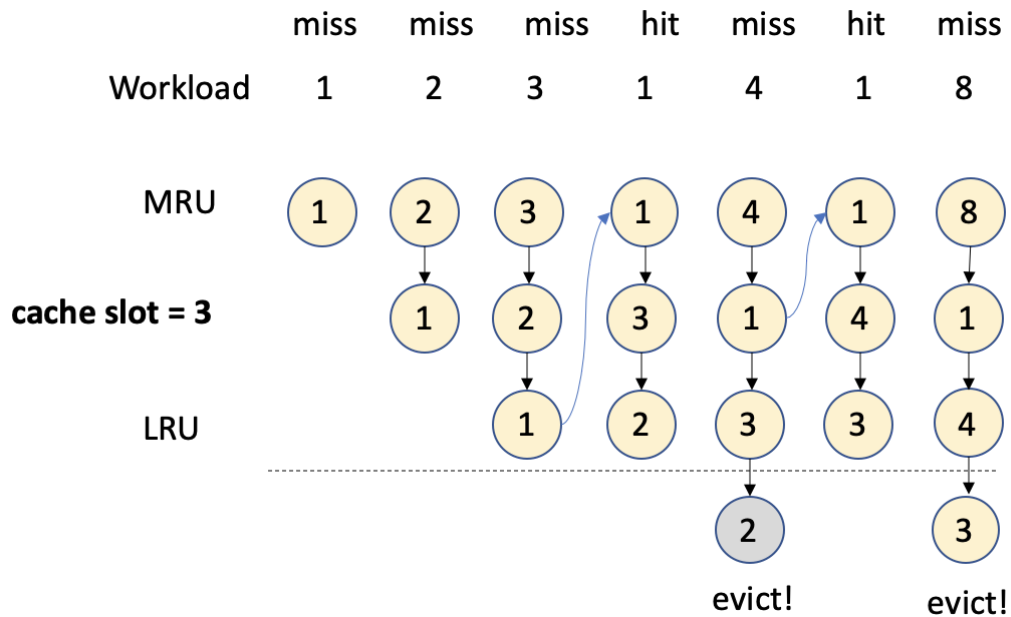
# 연결리스트로 LRU 시뮬레이터 구현하기

### 1. 개요

LRU(Least Recently Used) 알고리즘은 컴퓨터 응용의 메모리 접근이 통상 시간 지역성을 갖는다는 특성을 이용해 캐쉬 히트율(cache hit ratio)을 높이기 위해 사용되는 대표적인 교체 정책 (Replacement Policy)이다. LRU 알고리즘은 많은 시스템에서 연결리스트를 이용해 구현된다. 본 과제에서는 직접 구현한 CircularDoublyLinkedList 클래스를 이용해 LRU 시뮬레이터를 만드는 프로젝트를 수행하고, 시간복잡도를 분석해 본다.

### 2. LRU 시뮬레이터

실제 저장장치를 접근하는 워크로드를 입력으로 받아 캐쉬 크기를 변화시키며 LRU 알고리즘 적용 시 히트율을 측정하는 시뮬레이터를 만든다.



### 3. 요구사항

- 1) 아래의 코드 템플릿을 확장하여 LRU 시뮬레이터를 완성한다 (CircularDoublyLinkedList 클래스를 사용할 것).  
[https://github.com/eunjicious/ds\\_2022.git](https://github.com/eunjicious/ds_2022.git)

- 2) 캐쉬 크기를 100 에서 1000까지 변화시키면서 linkbench.trc 워크로드를 주입한 실행결과를 출력한다.

```
cache_slot = 100 cache_hit = 442240 hit ratio = 0.16915674079514073
cache_slot = 200 ...
```

- 3) LRU 시뮬레이터의 시간복잡도를 분석한다.
- 4) LRU 시뮬레이터의 시간복잡도를  $O(1)$ 으로 개선할 수 있는 방안을 제안한다.

#### 4. 제출방법

- 1) 제출기한: **4/20(수) 23:59** 까지 (지연제출 허용하지 않음. 코로나 확진 경우도 미허용. 기간 충분함.)
- 2) 제출물: 보고서 (요구사항 각각에 대한 결과물 캡처, 설명, 필요한 내용을 작성하여 제출)

**\*\* 결과에 대한 문의가 많아 제가 작성한 프로그램 결과 추가적으로 공지합니다 \*\***

혹시 다른 결과가 나오는 경우 본인 코딩이 맞다고 생각하면 생각대로 제출하면 됩니다.

```
cache_slot = 100 cache_hit = 943 hit ratio = 0.0943
cache_slot = 200 cache_hit = 1031 hit ratio = 0.1031
cache_slot = 300 cache_hit = 1101 hit ratio = 0.1101
cache_slot = 400 cache_hit = 1154 hit ratio = 0.1154
cache_slot = 500 cache_hit = 1236 hit ratio = 0.1236
cache_slot = 600 cache_hit = 1387 hit ratio = 0.1387
cache_slot = 700 cache_hit = 1501 hit ratio = 0.1501
cache_slot = 800 cache_hit = 1554 hit ratio = 0.1554
cache_slot = 900 cache_hit = 1813 hit ratio = 0.1813
```