2 Reverse Engineering Caches [60 points]

You are trying to reverse-engineer the characteristics of a cache in a system, so that you can design a more efficient, machine-specific implementation of an algorithm you are working on. To do so, you have come up with three patterns that access various *bytes* in the system in an attempt to determine the following four cache characteristics:

- Cache block size (8, 16, 32, 64, or 128 B)
- Cache associativity (1-, 2-, 4-, or 8-way)
- Cache size (4 or 8 KB)
- Cache replacement policy (LRU or FIFO)

However, the only statistic that you can collect on this system is *cache hit rate* after performing the access pattern. Here is what you observe:

Sequence	Addresses Accessed (Oldest → Youngest)								Hit Rate
1.	0	4	8	16	64	128			1/2
2.	31	8192	63	16384	4096	8192	64	16384	5/8
3.	32768	0	129	1024	3072	8192			1/3

Assume that the cache is initially empty at the beginning of the first sequence, but not at the beginning of the second and third sequences. The sequences are executed back-to-back, i.e., no other accesses take place between the three sequences. Thus, **at the beginning of the second (third) sequence, the contents are the same as at the end of the first (second) sequence**.

Based on what you observe, what are the following characteristics of the cache? Explain to get points.

(a) [15 points] Cache block size (8, 16, 32, 64, or 128 B)?

64 B.

Explanation:
Cache hit rate is 1/2 in sequence 1. This means that there are 3 hits, which are necessarily in addresses 4, 8, and 16. Thus, cache block size might be 32 or 64 B.

In sequence 2, there are only three misses, which are in addresses 8192, 16384, and 4096. The remaining 5 accesses are hits. For 63 to be a hit, the cache block size should be 64 B.

(b) [15 points] Cache associativity (1-, 2-, 4-, or 8-way)?

4-way.

Explanation:
We already know that the cache block size is 64 B. Thus, there are 6 offset bits.

If 1-way, 63 would miss, since 8192 would map to the same set regardless of cache size (i.e., bits 6 to 12 are equal).

Addresses 0, 4096, 8192, 16384, and 32768 map to the same set. If 2-way, the second access to 8192 and 16384 (in sequence 2) would not hit.

If 8-way, 1024 and 3072 would map to the same set as 0, 4096, 8192, 16384, and 32768, since they all share bits 6 to 9. In that case, 0 and 8192 would be both hits in sequence 3. This is not possible because there are only two hits in sequence 3. 32768, 1024, and 3072 are compulsory misses, while 129 is a hit (address 128 was accessed by sequence 1). Thus, either 0 or 8192 should miss in sequence 3.

(c) [15 points] Cache size (4 or 8 KB)?

8 KB.

Explanation:
We know that the cache is 4-way associative. The access to address 0 in sequence 3 is a miss, because the cache block was replaced by address 32768. Thus, access to 8192 should be a hit.

If 4-way and 4 KB, 1024 and 3072 would map to the same set as 8192. In that case, 8192 would miss. So the size of the cache should be 8 KB.

(d) [15 points] Cache replacement policy (LRU or FIFO)?

LRU.

Explanation:
For 8192 to hit in sequence 3, 4096 should have been replaced by 0. So the replacement policy is LRU, because FIFO would have replaced 8192.