Homework 2 Solutions
CS 336

Pranav Tikkawar

November 23, 2025

Problem 1: Linear Hashing

Problem Statement:
Consider a hash table that operates under linear hashing. When it started, the initial hashing function was $h_0(k) = k \bmod 4$; the hash table had 4 buckets (0,1,2,3), and the split pointer was $s = 0$.

(a) Table with 6 Buckets (0-5)

1. **Where is the split pointer?**

   - Start: 4 buckets (0-3), using h_0
   - Split 0 $\rightarrow$ add bucket 4, $s = 1$
   - Split 1 $\rightarrow$ add bucket 5, $s = 2$
   The split pointer is now at bucket **2**.

2. **How many hash functions are active?**

   - Buckets 0 and 1 have split using h_1. - Buckets 2 and 3 not yet split, still using h_0. - Buckets 4 and 5 resulted from the splits (also h_1).

   Thus, **2 hash functions** are active.

3. **Which hash functions? Which buckets use which?**

Bucket(s)	Hash Function
2, 3	$h_0(k) = k \bmod 4$
0, 1, 4, 5	$h_1(k) = k \bmod 8$

   So buckets 2 and 3 use h_0, while 0, 1, 4, 5 use h_1.

(b) Table with 32 Buckets (0-31)

1. **Where is the split pointer?**

   - 32 buckets = 2^5
   - Starting with 4 buckets: $4 \rightarrow 8 \rightarrow 16 \rightarrow 32$
   - Each time a full round of splits is complete, the pointer resets to 0.
   So after reaching 32, the split pointer is at **0**.

2. **How many hash functions are active?**

   - After a full round of splits, all buckets use the next hash function. - So, only **1 hash function** is active.

3. **Which hash function is active?**

   - $32 = 2^5$, so $h_3(k) = k \bmod 32$

   So, the active function is $h_3(k) = k \bmod 32$.

Problem 2: LRU Buffer Pool Management

Problem Statement:
A buffer pool holds 4 pages at a time. When a new page is requested and the pool is full, the Least Recently Used (LRU) policy evicts the page that hasn't been used for the longest time. Record changes after each operation.

Initial Buffer State

Frame	Page ID	Last Used
1	12	2
2	5	4
3	7	3
4	3	1

Start at time = 5 (most recent access was at 4).

Step-by-Step Operations

1. **Access Page 6 (time = 5)**

   - Page 6 not in buffer.
   - LRU: Page 3 (Last Used = 1, Frame 4).
   - Evict Page 3, bring in 6 (Last Used = 5).

Frame	Page ID	Last Used
1	12	2
2	5	4
3	7	3
4	6	5

2. **Access Page 3 (time = 6)**

   - Page 3 not in buffer.
   - LRU: Page 12 (Last Used = 2, Frame 1).
   - Evict Page 12, bring in 3 (Last Used = 6).

Frame	Page ID	Last Used
1	3	6
2	5	4
3	7	3
4	6	5

3. **Access Page 5 (time = 7)**

   - Page 5 is in buffer (Frame 2).
   - Just update Last Used to 7.

Frame	Page ID	Last Used
1	3	6
2	5	7
3	7	3
4	6	5

4. **Access Page 8 (time = 8)**
   - Page 8 not in buffer.
   - LRU: Page 7 (Last Used = 3, Frame 3).
   - Evict Page 7, bring in 8 (Last Used = 8).

Frame	Page ID	Last Used
1	3	6
2	5	7
3	8	8
4	6	5

5. **Access Page 5 (time = 9)**
   - Page 5 is in buffer (Frame 2).
   - Just update Last Used to 9.

Frame	Page ID	Last Used
1	3	6
2	5	9
3	8	8
4	6	5