
1a. Hash Table with Minimum and Maximum Blocks

Suppose we have a file of 2,500,000 records that we want to hash into a table with 2,500 buckets. Each block can store 200 records, and we wish to maximize block utilization while ensuring that no two buckets share a block. Empty buckets do not consume a block. What is the minimum and maximum number of blocks required to store this hash table?

1b. Hash Table with Variable Records

Suppose we have a file of 750,000 records that needs to be hashed into a table with 1,500 buckets. Each block can hold 150 records. We wish to minimize the number of blocks used while ensuring that no two buckets share a block. Empty buckets do not consume a block. Determine the minimum and maximum number of blocks required.

2a. Extensible Hashing Simulation

Suppose keys are hashed to five-bit sequences, and blocks can hold four records. If we start with a hash table with two empty blocks (corresponding to 0 and 1), show how the hash table evolves when we insert records with the following hash values:
00000, 00001, ..., 11111, and the method of hashing is extensible hashing.

2b. Simulating Hash Table Growth

Suppose keys are hashed to three-bit sequences, and each block can hold two records. Begin with a hash table consisting of two empty blocks (corresponding to 0 and 1). Trace the changes in the hash table as you insert keys with the following hash values:
000, 001, ..., 111, using extensible hashing. Show the final state of the hash table.

3a. Recursive Overflow Probability

In an extensible hash table, blocks can hold n records before splitting. If a split occurs after n+2 records are added, what is the probability that an overflow will have to be handled recursively, i.e., all records are placed into the same one of the two blocks created during the split?

3b. Recursive Overflow Probability with Additional Records

Consider an extensible hash table with blocks capable of holding n records. If a split occurs after n+4 records are added, calculate the probability that the overflow requires recursive handling, i.e., all n+4 records are assigned to the same one of the two newly created blocks.

Exam.p.

"A linear hash table with a capacity threshold of 80%. Suppose that keys are hashed to four-bit blocks can hold two records.
What is the minimum number of records we have to insert to get a hash table with 5 buckets?
Provide an example of a hash table with exactly 4 buckets and 2 overflow blocks. Draw the n with their hash values."

---

two more examples

Example 1:
A linear hash table has a capacity threshold of 75%. Assume that keys are hashed to five-bit values and that each bucket can store three records.

1. What is the minimum number of records that must be inserted to ensure the hash table expands to 6 buckets?
2. Provide an example of a hash table with exactly 5 buckets and 3 overflow blocks. Draw the records along with their hash values.

Example 2:
A dynamic hash table follows a capacity threshold of 85%. Suppose keys are hashed to six-bit values and each bucket can hold four records.

1. What is the minimum number of records needed to expand the hash table to 8 buckets?
2. Construct an example of a hash table with exactly 6 buckets and 4 overflow blocks. Illustrate the records and their hash values.

Question 3a:

In an extensible hash table with **2 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split? Assume that the hash function distributes keys uniformly.

Question 3b:

In an extensible hash table with **3 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split? Assume that the hash function distributes keys uniformly.

Question 3c:

In an extensible hash table with **4 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split? Assume that the hash function distributes keys uniformly.

Key Concepts:

1. **Extensible Hashing:**
   - Extensible hashing is a dynamic hashing technique where the hash table grows by doubling the number of buckets when a bucket overflows.
   - Each bucket can hold a fixed number of records (e.g., 2, 3, or 4 records per block).
2. **Overflow Handling:**
   - When a bucket overflows, it is split into two buckets, and the records are redistributed based on an additional bit of the hash value.
   - If all records in the overflowing bucket hash to the same new bucket, the overflow must be handled recursively.
3. **Probability Calculation:**
   - The probability that all records in the overflowing bucket hash to the same new bucket depends on the number of records in the block and the uniformity of the hash function.

How to Solve These Questions:

Step 1: Understand the Problem

- For a block with n*n* records, an overflow occurs when the block is full.
- When the block is split, each record has a 50% chance of going to either of the two new buckets (assuming a uniform hash function).

Step 2: Calculate the Probability

- The probability that all n records go to the same new bucket is:

$$P = \left(\frac{1}{2}\right)^{n-1}$$

  - This is because the first record can go to either bucket, and each subsequent record must match the first record's bucket choice.

Step 3: Apply the Formula

- For **Question 3a** ($n = 2$):

$$P = \left(\frac{1}{2}\right)^{2-1} = \frac{1}{2}$$

- For **Question 3b** ($n = 3$):

$$P = \left(\frac{1}{2}\right)^{3-1} = \frac{1}{4}$$

- For **Question 3c** ($n = 4$):

$$P = \left(\frac{1}{2}\right)^{4-1} = \frac{1}{8}$$

Final Answers:

1. **Question 3a:** The probability is $\frac{1}{2}$.
2. **Question 3b:** The probability is $\frac{1}{4}$.
3. **Question 3c:** The probability is $\frac{1}{8}$.

Explanation of the Formula:

- The formula $P = \left(\frac{1}{2}\right)^{n-1}$ arises because:
  - The first record can go to either bucket (probability 1).
  - Each subsequent record must match the first record's bucket choice (probability $\frac{1}{2}$ for each record).
  - Since there are $n-1$ subsequent records, the total probability is $\left(\frac{1}{2}\right)^{n-1}$.

Question 3a:

In an extensible hash table with **2 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split?

Solution:
We split a block when we have **3 records** for this block (since $n = 2$). When we split a block, we consider an additional bit to decide to which of the two blocks a record will be assigned. When all **3 records** go into the same block, they have the same value for this bit. The value of this bit can be for all records either **0** (with a probability $\left(\frac{1}{2}\right)^3$) or **1** (also with a probability $\left(\frac{1}{2}\right)^3$). In total, the probability is:

$$P = \left(\frac{1}{2}\right)^2 = \frac{1}{4}.$$

Question 3b:

In an extensible hash table with **3 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split?

Solution:
We split a block when we have **4 records** for this block (since $n = 3$). When we split a block, we consider an additional bit to decide to which of the two blocks a record will be assigned. When all **4 records** go into the same block, they have the same value for this bit. The value of this bit can be for all records either **0** (with a probability $\left(\frac{1}{2}\right)^4$) or **1** (also with a probability $\left(\frac{1}{2}\right)^4$). In total, the probability is:

$$P = \left(\frac{1}{2}\right)^3 = \frac{1}{8}.$$

Question 3c:

In an extensible hash table with **4 records per block**, what is the probability that an overflow will have to be handled recursively, i.e., all members of the block will go into the same one of the two blocks created in the split?

Solution:
We split a block when we have **5 records** for this block (since $n = 4$). When we split a block, we consider an additional bit to decide to which of the two blocks a record will be assigned. When all **5 records** go into the same block, they have the same value for this bit. The value of this bit can be for all records either **0** (with a probability $\left(\frac{1}{2}\right)^5$) or **1** (also with a probability $\left(\frac{1}{2}\right)^5$). In total, the probability is:

$$P = \left(\frac{1}{2}\right)^4 = \frac{1}{16}.$$

1a) Suppose we have a file of 2,000,000 records that we want to hash into a table with 2,000 buckets. Each bucket can hold 50 records, and we wish to minimize the blocks used while not allowing two buckets to share a block. Empty buckets do not consume a block. What is the minimum and maximum number of blocks needed?

Solution

- **Best Case:**
  Each bucket is completely full and contains $\frac{2{,}000{,}000 \text{ records}}{2{,}000 \text{ buckets}} = 1,000 \text{ records}$.
  Each bucket requires:

$$\lceil \frac{1,000}{50} \rceil = 20 \text{ blocks.}$$

  Total blocks:

$$2,000 \text{ buckets} \times 20 \text{ blocks/bucket} = 40,000 \text{ blocks.}$$

- **Worst Case:**
  $1,999$ buckets contain one record each. Each bucket uses 1 block.
  The last bucket contains the remaining $2,000,000 - 1,999 = 1,998,001$ records, which requires:

$$\lceil \frac{1,998,001}{50} \rceil = 39,961 \text{ blocks.}$$

  Total blocks:

$$1,999 \text{ blocks} + 39,961 \text{ blocks} = 41,960 \text{ blocks.}$$

Answer:

- Minimum: **40,000 blocks**
- Maximum: **41,960 blocks**

Question 3a:
In an extensible hash table with n records per block, what is the probability that an overflow will occur such that all n+1 records have hash values that assign them to the same block after splitting? Assume the hash function produces uniform random values.

Explanation:
When a block overflows in an extensible hash table, it splits into two new blocks, and an additional bit of the hash value determines which block each record goes into. The probability of all n+1 records ending up in the same block is given by $(1/2)^n$, as the split evenly assigns records to one of the two blocks with a 50% chance per record. This probability reflects the chance of requiring further recursive splits due to an imbalance.

Question 3b:
Suppose we have an extensible hash table with n records per block, and each record's hash value is independently distributed. What is the probability that, after a split, all but one of the n+1 records end up in the same block, requiring additional reorganization?

Explanation:
In this case, the overflow condition is slightly relaxed, where n records stay in one block, and only one record moves to the new block. For n+1 records, this can happen in n+1 possible ways (choosing which record moves), and each configuration happens with probability $(1/2)^{n+1}$. Thus, the total probability of this specific type of imbalance is $(n+1) \cdot (1/2)^{n+1}$.

Concept Explanation:
The probability of recursive overflow in extensible hashing is rooted in the behavior of the hash function. Extensible hashing dynamically adjusts the structure of the hash table, ensuring efficiency while managing overflow. When a block overflows, splitting redistributes records based on an additional hash bit. However, there is a small chance that the split will fail to evenly distribute records, leading to recursive splits or further adjustments. This probability diminishes exponentially as the number of records n in a block increases.