Weekplan: Hashing

Philip Bille

References and Reading

[1] Notes from Aarhus, Peter Bro Miltersen.
[2] Scribe notes from MIT.


Exercises

1 Basic Probability Theory Refresh Bonus In case your knowledge of probability theory is rusty. Solve the following self-help exercises.

1.1 Prove linearity of expectation.

1.2 Prove that the expectation of the indicator function for \( h(x) = h(y) \) (1 if \( h(x) = h(y) \) and 0 otherwise) is equal to the probability that \( h(x) = h(y) \).

1.3 Show that the expected number of trials to get a perfect hashing function using an array of size \( n^2 \) is \( \leq 2 \).

2 Streaming Statistics An IT-security friend of yours wants a high-speed algorithm to count the number of distinct incoming IP-addresses in his router to help detect denial of service attacks. Can you help him?

3 Dense Set Hashing A set \( S \subseteq U = \{0, \ldots, u - 1\} \) is called dense if \( |S| = \Theta(u) \). Suggest a simple and efficient dictionary data structure for dense sets.

4 Multi-Set Hashing A multi-set is a set \( M \), where each element may occur multiple times. Design an efficient data structure supporting the following operations:

- \( \text{add}(x) \): Add an(other) occurrence of \( x \) to \( M \).
- \( \text{remove}(x) \): Remove an occurrence of \( x \) from \( M \). If \( x \) does not occur in \( M \) do nothing.
- \( \text{report}(x) \): Return the number of occurrences of \( x \).

5 Linear Space Hashing The chained hashing solution for the dynamic dictionary problem presented assume that \( m = \Theta(n) \). Solve the following exercises.

5.1 What is the space complexity of chained hashing without this assumption?

5.2 Give a solution that achieves \( O(n) \) space and the same time complexities without assuming \( m = \Theta(n) \). Hint: Think dynamic tables.
6 Graph Adjacency  Let $G$ be a graph with $n$ vertices and $m$ edges. We want to represent $G$ efficiently and support the following operation.

- $\text{adjacent}(v, w)$: Return true if nodes $v$ and $w$ are adjacent and false otherwise.

Solve the following exercises:

6.1 Analyse the space and query time in terms of $n$ and $m$ for the classic adjacency matrix and adjacency list representation.

6.2 Design a data structure that improves both the adjacency matrix and adjacency list.

7 Lost Integer Puzzles  Suppose that you receive a stream of $n - 1$ distinct integers from the set $\{1, \ldots, n\}$, i.e., the stream consists of all of $\{1, \ldots, n\}$ except a single missing integer. We want a space-efficient algorithm that efficiently computes this integer during a single pass over the input stream. Solve the following exercises:

7.1 Show how to find the lost integer using $O(n)$ space.

7.2 [$\ast$] Show how to find the lost integer using $O(1)$ space.

7.3 [$\ast\ast$] Suppose there are now two lost integers. Show how to find them using $O(1)$ space.