**REPORT**

A screenshot of a computer

Description automatically generated

Analysis of Results:

Performance Differences

1. 1D Hash Table:
   * Consistently required fewer spots to be checked during insertion, removal, and search.
   * This result is expected because it uses linear probing, which provides continuous probing across the table when collisions occur, allowing for fewer iterations in most cases.
2. 2D Hash Table:
   * Required more spots to be checked in all phases.
   * This is due to its design: collisions are handled within a single row, which is limited to 5 slots. If the row is full, insertion fails, or multiple attempts are needed, increasing the spots checked.

Situational Advantages and Drawbacks

1. 1D Hash Table:
   * Ideal Situations:
     + When a large number of slots are available, and the table is sparsely populated, linear probing efficiently finds empty slots.
   * Least Ideal Situations:
     + As the table becomes more crowded, clustering may occur, increasing the number of spots checked during insertion or search.
2. 2D Hash Table:
   * Ideal Situations:
     + When data distribution is relatively uniform, and rows have space available for collision handling.
     + Can be useful for applications where grouping data by rows or categories is meaningful.
   * Least Ideal Situations:
     + If hash collisions frequently place items in the same row, rows can fill up quickly, leading to failed insertions and increased probes.
     + Less efficient for sparse tables compared to 1D hash tables.

Did the Results Meet Expectations?

The results align with theoretical expectations:

1. 1D Hash Table is more efficient overall due to its linear probing, which provides a simple and effective collision resolution mechanism across the full range of slots.
2. 2D Hash Table incurs more overhead because of its constrained row design and localized collision handling.

Which Performed Best?

* 1D Hash Table consistently performed better in terms of fewer spots checked for all operationThis result is expected because the 1D hash table has a larger space available for collision resolution, while the 2D hash table is constrained by the 5-slot-per-row limitation.

Recommendations:

* Use 1D Hash Table for applications where efficiency and simplicity are key, especially with larger datasets or when the table is expected to become densely populated.
* Consider 2D Hash Table for scenarios where organizing data into smaller, fixed-size rows is advantageous and the dataset size aligns with the row/slot constraints.