Semester VI (B.Tech)

Jaypee University of Engineering and Technology, Guna

Advanced Programming Lab-3 18B17CI673

Lab Exercise 9: Hash Tables

Title: Performance Comparison of Hash Table Implementations

Mode: Self Learning

Outcomes:

1. To compare the performance of hash tables with different collision resolution methods and load factors.
2. To understand the trade-offs between hash table implementations and their impact on the performance.
3. To gain hands-on experience in implementing hash tables in a programming language.

Methodology:

1. Selection of programming language and libraries: Python and NumPy
2. Selection of collision resolution methods: chaining, linear probing, and quadratic probing
3. Generation of random inputs with different sizes and distributions using NumPy
4. Implementation of the selected hash table implementations using Python and NumPy
5. Running the implemented hash tables on the generated inputs
6. Measuring the running time and memory usage of the hash tables using appropriate tools such as the Python time module and memory_profiler library
7. Analysis of the results and comparison of the hash tables based on their performance using appropriate statistical analysis techniques
8. Writing a report summarizing the methodology, results, and conclusions of the experiment.

Steps:

1. Implement a hash table with chaining collision resolution.
2. Implement a hash table with linear probing collision resolution.
3. Implement a hash table with quadratic probing collision resolution.
4. Generate random inputs with different sizes and distributions using NumPy.
5. Insert the generated key-value pairs into the implemented hash tables.
6. Search for random keys in the implemented hash tables and measure the running time and memory usage.
7. Analyze the results and compare the hash tables based on their performance using appropriate statistical analysis techniques.
8. Repeat steps 4-7 for different load factors, such as 50%, 75%, and 90%.
9. Write a report summarizing the methodology, results, and conclusions of the experiment.

Experiments:

1. Write a program in your chosen programming language to implement a hash table with chaining collision resolution. The hash table should be able to insert and retrieve key-value pairs.
2. Generate a list of random key-value pairs with different key sizes and distributions. You can use a library like NumPy to generate random numbers.
3. Insert the key-value pairs into the hash table.
4. Implement a function that searches for a key in the hash table and returns the corresponding value