***Name:*** Muhammad Wasiq ***Roll no:*** 03-134222-082

|  |
| --- |
| **LINEAR PROBING** |

***INSERTION***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 423 | 344 | 345 |
| 425 | 339 | 331 |
| 430 | 323 | 320 |

***SEARCHING***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 78 | 67 | 69 |
| 76 | 84 | 79 |
| 106 | 95 | 91 |

|  |
| --- |
| **LINKLIST CHANNING** |

***INSERTION***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 445 | 226 | 222 |
| 443 | 226 | 227 |
| 438 | 222 | 222 |

***SEARCHING***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 92 | 48 | 44 |
| 73 | 47 | 42 |
| 91 | 49 | 49 |

|  |
| --- |
| **BST CHANNING** |

***INSERTION***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 437 | 216 | 220 |
| 458 | 214 | 213 |
| 442 | 230 | 231 |

***SEARCHING***

|  |  |  |
| --- | --- | --- |
| **PRIME NUMBERS** | **EVEN NUMBERS** | **ODDNUMBERS** |
| 141 | 121 | 119 |
| 122 | 115 | 120 |
| 142 | 126 | 127 |

Collisions:

**Average Insertion:**

1. Linear Probing:

- DATA1: 426

- DATA2: 332

- DATA3: 335

2. Linked-list Chaining:

- DATA1: 442

- DATA2: 223

- DATA3: 224

3. BST Chaining:

- DATA1: 445

- DATA2: 221

- DATA3: 220

**Average Searching:**

1. Linear Probing:

- DATA1: 86

- DATA2: 79

- DATA3: 82

2. Linked-list Chaining:

- DATA1: 85

- DATA2: 45

- DATA3: 48

3. BST Chaining:

- DATA1: 135

- DATA2: 122

- DATA3: 120

**Conclusion:**

1. Linked-list Chaining consistently shows lower average insertion and searching times compared to both Linear Probing and BST Chaining.
2. Linear Probing is generally faster than BST Chaining but slower than Linked-list Chaining in terms of both insertion and searching.
3. BST Chaining tends to have higher average searching times, making it less efficient for search operations compared to the other two techniques.

**Reasons:**

1. Memory Utilization:

* Linked-list Chaining: Offers good memory utilization as it dynamically allocates memory for each element.
* Linear Probing: May lead to clustering, reducing memory efficiency.
* BST Chaining: Requires additional memory for pointers and tree structure.

2. Collision Handling:

* Linked-list Chaining: Handles collisions efficiently by chaining elements in linked lists.
* Linear Probing: Can suffer from clustering and increased collisions.
* BST Chaining: The tree structure may become unbalanced, affecting performance.

3. Search Efficiency:

* Linked-list Chaining: Provides constant time complexity for searching in the average case.
* Linear Probing: May suffer from clustering and probing, affecting search efficiency.
* BST Chaining: Depends on the balance of the tree, and an unbalanced tree can result in longer search times.

In summary, based on the average collisions, Linked-list Chaining is the preferred technique due to its better performance in terms of both average insertion and searching times, as well as efficient collision handling. (in case of 1000 nodes).