**Lab 8: Hash functions and Hash Table**

**Question 1:** Implement the double hash function. You should implement only **put** function in the HashTable.java file.

**DOUBLE\_HASH** (key, value)

1. j = 0
2. **while** j < N:
3. h\_k = hashFunction\_1(key) % 13
4. d\_k = hashFunction\_2(key) % 13
5. probe = h\_k+ j \* d\_k
6. item = table[probe]
7. **if** item == **then**
8. insert new Entry < key, value>
9. **else if** item.key == key **then**
10. table[idx].value = value
12. j++;

**Question 2**: There are many functions used for collision handling in hashing. Explain one of them using the example below (Except the double hashing). Show all steps (**N = 13**).

Input: [18, 41, 22, 44, 59, 32, 31, 73]

**Question 3:** What is a good Hash function? Which factors affect the performance of the hash function? Explain each factor.

**Question 4:** Which implementation is better? HashTable LinkedList implementation or Array implementation? Evaluate implementation for search, remove and insert operations.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **search** | **insert** | **remove** |
| Array |  |  |  |
| LinkedList |  |  |  |

**Question 5:**  Fill in the following table with the worst-case efficiency classes using Θ notation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **search** | **insert** | **remove** |
| AVL Tree |  |  |  |
| Hash Table |  |  |  |
| Binary Search Tree |  |  |  |