Cpts 223-HW5

Yurun Han

04/11/21

**1.**

**a)**

hash (12) = (12 \* 12 + 3) % 11 = 4

hash (9) = (9 \* 9 + 3) % 11 = 7

hash (3) = (3 \* 3 + 3) % 11 = 1

hash (0) = (0 \* 0 + 3) % 11 = 3

hash (42) = (42 \* 42 + 3) % 11 = 7

hash (98) = (98 \* 98 + 3) % 11 = 4

hash (70) = (70 \* 70 + 3) % 11 = 8

hash (1) = (1 \* 1 + 3) % 11 = 4

Separate Chaining (buckets)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 |  | 0 | 12 |  |  | 9 | 70 |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Linear Probing: probe(i’) = (i + 1) % TableSize

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 |  | 0 | 12 | 98 | 1 | 9 | 42 | 70 |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Quadratic Probing: probe(i’) = (i \* i + 5) % TableSize

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 42 |  | 0 | 12 | 3 |  | 9 | 70 | 98 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**b)**

load factor = 53491/100001 = 0.5349

**2.**

|  |  |
| --- | --- |
| Function | Big-O Complexity |
| Insert(x) | O(1) |
| Rehash(x) | O(n) |
| Remove(x) | O(1) |
| Contains(x) | O(1) |

**3.**

int hashit (int key, int tablesize )

{  
int ret=key % tablesize;  
return ret;  
}

int hashit (string key, int tablesize )

{  
int hash, i;  
for (hash = key.length(), i = 0; i < key.length(); i++)  
hash += key.charAt(i);  
return (hash % tablesize);

}

**4.**

Parallel computing is a type of calculation that allows multiple tasks to run at the same time. And you can divide the problem into small problems and proceed at the same time. Multiple threads can be scheduled to the CPU for execution.

**5.**

Task parallelism and Data parallelism.