

Hashing Notes 1

February 26, 2017

1 Algorithms and Data Structures: Hashing

1.1 Worst case complexity of hashing is $O(n)$. But, on an average it gives $O(1)$.

Direct Addressing: One unique location for every possible key.

1.2 Load Factor:

Number of items stored in a hash table / size of the table

This tells whether the hash function is distributing the keys evenly in the table or not. How? well, the load factor tells the average size of chains that would be built in the table as a result of chaining. Now, if actual size of a chain corresponding to a certain key goes far beyond the load factor, then we would know that the hash function is not distributing the keys evenly.

1.3 Collision resolution techniques:

1. Separate Chaining
2. Open Addressing or Closed Hashing. Collision resolution technique is called "probing" in this method.
 1. Linear Probing: Rehash : $\text{rehash}(\text{key}) = (n+1) \% \text{tablesize}$. Interval between probes is fixed at 1. Problem of Clustering is seen.
 2. Quadratic Probing: Rehash:

$$\text{rehash}(\text{key}) = (n + k^2) \bmod (\text{tablesize})$$

. Intervals between probes increases proportionally to the hash value. Problem of secondary clustering is seen.

3. Double Hashing

1. Interval between probes is computed by another hash function.
2. $h_2(\text{key}) \neq 0$. $h_2 \neq h_1$

Hashing is useful if the data is static and has unique keys.

If the problem domain needs more search operations than insert operations.

2 Bloom Filter

A probabilistic data structure which was designed to check whether an element is present in a set with memory and time efficiency.

It tells whether the element is definitely not there in the set or might be there in the set.