Report

CSE-208
(Data Structures And Algorithms – II)

Offline on Hashing

Submitted by:

Tanjeem Azwad Zaman

Roll: 1805006

Dept. of CSE, BUET

Section: A Batch:18

Date of Submission:

19.01.2022

Hashing methods used:

1. ***Polynomial Rolling Hash Function:***

$$\begin{aligned}\text{hash}(s) &= s[0] + s[1] \cdot p + s[2] \cdot p^2 + \ldots + s[n-1] \cdot p^{n-1} \mod m \\ &= \sum_{i=0}^{n-1} s[i] \cdot p^i \mod m,\end{aligned}$$

Where,

- s = string of length n;
- s[i] = (int)(s[i]) – (int) (a) + 1
- eg. [a -> 1, b->2 …. z ->26]
- p is a prime roughly equal to the number of alphabets (like 31)
- m is a large prime (our table size was taken as a prime, so m = table size)

2. ***DJB2:***

This is a function defined by Dan Bernstein. It's recursive definition is:

h(0) = 0;
h(i) = (h(i-1) * 33) + s[i] [for I = 1 to n-1]

Specific Definitions:

- **Hash1()** was a Polynomial Rolling Hash Function
  with p = 31, m = table size = 10007 (usually)
- **Hash2()** was a DJB2 Hashing function
- **auxHash()** was also a Polynomial Rolling Hash Function
  with p = 29, m = table size = 10007 (usually)

Furthermore,
in **Double hashing** [*doubleHash(k, i) = (Hash(k) + i × auxHash(k)) % N*]
and in
Custom Hashing [*customHash(k, i) = (Hash(k) + C1 × i × auxHash(k) + C2 × i 2) % N*]

- **N** was taken to be the table size (10007 usually).
- **C1** was taken to be 43 (a prime)
- **C2** was taken to be 47 (another Prime)

TABLES:

Table 1: *(Word count = **10000**, Table size = **10007**, searched = **1000**)*

	Hash 1		Hash 2	
	# of Collisions	Avg. Probes	# of Collisions	Avg. Probes
Chaining Method	3669	1.503	3678	1.497
Double Hashing	59837	6.474	57439	5.879
Custom Probing	64440	6.124	64896	5.937

Table 2: *(Word count = **10000**, Table size = **20029**, searched = **1000**)*

	Hash 1		Hash 2	
	# of Collisions	Avg. Probes	# of Collisions	Avg. Probes
Chaining Method	2120	1.262	2167	1.25
Double Hashing	3862	1.409	3954	1.358
Custom Probing	3877	1.408	3917	1.351

Table 3: *(Word count =* ***10000****, Table size =* ***100000****, searched =* ***1000****)*

	Hash 1		Hash 2	
	# of Collisions	Avg. Probes	# of Collisions	Avg. Probes
Chaining Method	468	1.053	473	1.046
Double Hashing	530	1.055	517	1.047
Custom Probing	526	1.056	521	1.051

Table 3: *(Word count =* ***10000****, Table size =* ***1000000****, searched =* ***1000****)*

	Hash 1		Hash 2	
	# of Collisions	Avg. Probes	# of Collisions	Avg. Probes
Chaining Method	47	1.008	43	1.003
Double Hashing	48	1.007	43	1.005
Custom Probing	45	1.008	46	1.004

Observations and Inferences:

- For smaller table sizes (of the order of Word count), Chaining method significantly out-performed the other two (since, after 1 hash, the end of the Doubly linked list could be reached in O(1))
- For the larger table sizes (> Word count x 10), the performances were almost identical, with negligible differences
- Whether table size was a prime, influenced the result greatly (for Double Hashing and Custom Probing) when Table size and Word count were of the same order. As table size increased, it mattered less.
- The hash functions yielded similar performances; rather the collision resolution techniques made significant differences at lower table sizes.