

Hashing Performance Analysis

1.

I have chosen this specific hash function because it executes so much faster when compared to other hash functions like **(sum of ascii of string) mod table size** and **(first letter ascii) mod table size**.

2.

Size	Performance of english_large.txt	Performance of english_small.txt
200000	12.642293930053711 seconds	0.38570165634155273 seconds
300000	0.9866938591003418 seconds	0.42940616607666016 seconds
400000	0.9639694690704346 seconds	0.5194253921508789 seconds

According to the above table, the larger the table size, slower the runtime will be.

Because there will be less collisions if the table size is larger, and the collisions slow down the runtime.

3.

I) **number of collisions** for **english_large.txt**

Size	Linear Probing	Quadratic Probing	Double Hashing
200000	22605601	756065	8182365
300000	249490	216042	155840
400000	165773	164598	107439

According to the above table, there are a smaller number of collisions for double hashing if we increase the table size. Also, there is no drastic difference between all 3 collision techniques if we increase the table size and there are less collisions too in this case.

II) **average probe length** for **english_large.txt**

Size	Linear Probing	Quadratic Probing	Double Hashing
200000	117.26421955120787	4.888563155431434	43.083211183286785
300000	2.2831669521120386	2.111138541297002	1.8015100317332964
400000	1.8525970385685557	1.8465538257394578	1.552575951613152

According to the above table, the average probe length is really good for quadratic probing for 200000 table size. Also, there is no drastic difference between all 3 collision techniques if we increase the table size and the average probe length is less too in this case.