Name: Ritika, Eunho, Tomer

Report

1. **How it works?**

HASHFUNCTION1

For this Hash function, we take the board which is a 2D array, and iterate through the first row each column and the last row each column, convert each character to ASCII values (including the spaces), and add them together. After we get the sum of the ASCII values of each character in the first and last row, we divide it by the number of slots in our hash table and only get the reminder part using the MOD function.

HASHFUNCTION2

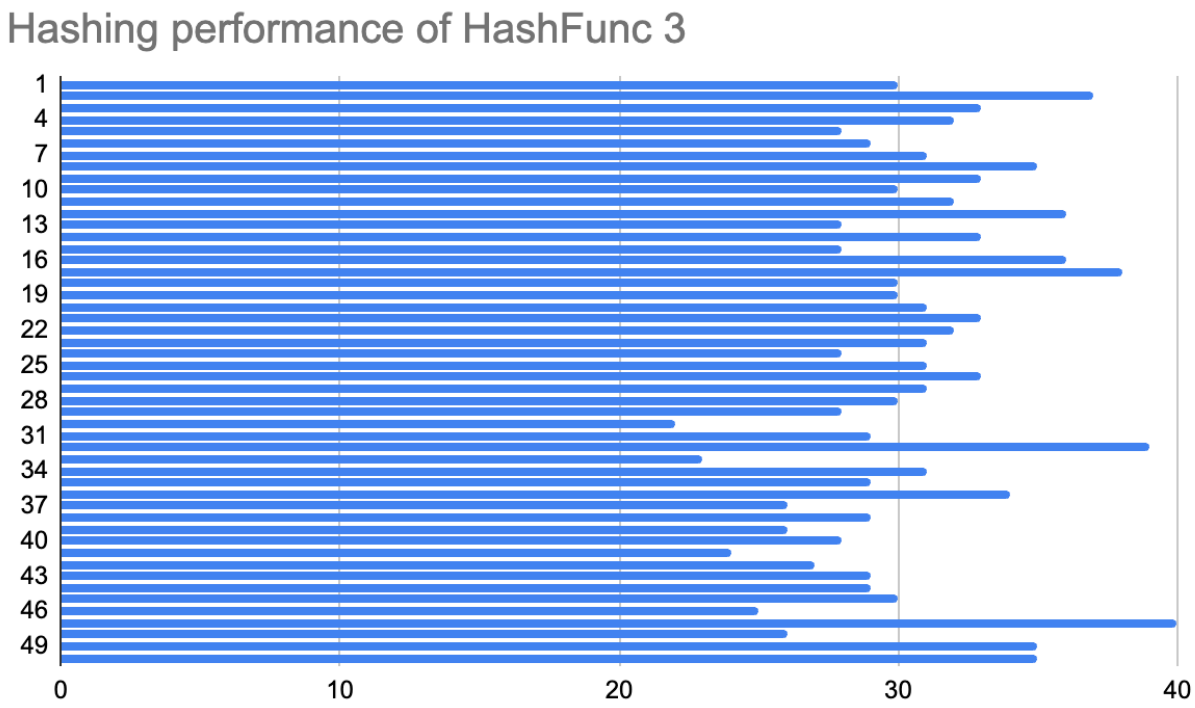
For this Hash function, we first iterate through all characters in the board which is a 2D array and, convert each character to ASCII values, and add it to the sum multiplied by a prime number, in this case, we choose “37”, to make it more independent of the actual data(character). And then we add the row index i to sum. After we iterate through all characters and compute the sum, then we divide it by 8, distribute it furthermore, and divide it by the number of slots in our hash table and only get the reminder part using the MOD function, which will be our slot number.

HASHFUNCTION3

For this Hash function, we first iterate through all characters in the board which is a 2D array, and, convert each character to ASCII values and divide it by its column index (column position in the board). Then we divide pi by i (row index) and multiply to sum for each character except the first character. Here we are trying to make the fractional part of the number as random as possible (unrelated to the data). Now, we separate the fractional part and integer part of the sum(value) we calculated and just extract the fractional part. Then, we multiply the fractional part by pi^8 and again convert it into an integer and divide it by the number of slots in our hash table and only get the reminder part using the MOD function, which will be our slot number.

2. **Bar chart (Best Hashing Performance)**

Best hashing performance for the Config2.txt file with 50 hash table slots, with the x-axis as the number of items in each slot of the 50 slots.



3. **Hash performance**

	Data File	Hash Size	Total Unique	Mean	Std	Min	Max
HASHFUNC1	Config1	10	23	2.3	4.64866	0	13
	Config2	50	1533	30.66	52.0863	0	173
HASHFUNC2	Config1	10	23	2.3	1.00499	1	4
	Config2	50	1533	30.66	9.77673	9	56
HASHFUNC3	Config1	10	23	2.3	1.1	1	4
	Config2	50	1533	30.66	3.89672	22	40