Hashing Matrices

Hash Table Size	Collision Resolution Method	Hash1 (Collisions)	Hash1 (Avg Probes)	Hash2 (Collisions)	Hash2 (Avg Probes)
5003	Separate Chaining	5684	1.89	5660	1.947
	Double Hashing	32114	3.662	31995	2.866
	Custom Probing	39968	5.12	34083	3.594
10007	Separate Chaining	3654	1.432	3625	1.465
	Double Hashing	57973	4.722	56510	5.893
	Custom Probing	62768	6.639	68919	7.537
20011	Separate Chaining	2140	1.22	2084	1.219
	Double Hashing	3807	1.363	3708	1.315
	Custom Probing	3854	1.352	3766	1.316

Hash Functions:

Hash1:

This function computes a hash value for a given string str using the hash algorithm based on the djb2 (Daniel J. Bernstein) hash function. The algorithm initializes a hash value called hash to 5381. Then, for each character c in the input string str, the hash value is updated using the following formula: hash = ((hash << 5) + hash) + c. This involves shifting the current hash value left by 5 bits and then adding the current hash value to it before adding the ASCII value of the current character c. Finally, the hash value is returned modulo M, where M is the desired range of hash values.

Hash2:

This function uses a hash algorithm based on the FNV-1a (Fowler-Noll-Vo) hash function. It initializes a hash value called hash to a specific constant, 14695981039346656037UL. For each character c in the input string str, the hash value is modified using the bitwise XOR operation (hash ^= c) followed by multiplication with a prime constant (hash *= 1099511628211UL). This process iterates through all the characters in the string. Finally, the hash value is returned modulo M.

auxHash:

This function computes a hash value for a given string s using a simple hash algorithm. It initializes a hash value called hash to 7 and a prime constant prime to 19. Then, for each character in the input string s, the hash value is updated using the formula: hash = (hash * prime + s[i]) % M. This involves multiplying the current hash value by the prime constant and adding the ASCII value of the current character. The resulting hash value is then taken modulo M. If the computed hash value modulo M equals 0, the function returns 1; otherwise, it returns the computed hash value modulo M.