© Universiti Tun Hussein Onn Malaysia Publisher's Office

JSCDM

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/jscdm

Journal of Soft Computing and Data Mining

Performance Evaluation of Quadratic Probing and Random Probing Algorithms in modeling Hashing Technique

Raghda Abdulbaqi Mugher[1*], Nafea Ali Majeed Alhammadi[2]

[1]Directorate of Education in Diyala,
Ministry of Education, 32001, Diyala, IRAQ

[2]Department of Computer Sciences,
Shatt Al-Arab University College, 61002, Basrah, IRAQ

*Corresponding Author

DOI: https://doi.org/10.30880/jscdm.2022.03.02.006
Received 03 October 2022; Accepted 26 October 2022; Available online 01 November 2022

Abstract: In hashing technique, a hash table and hash map represent a data structure for a group of objects to map between key and value pairs, as the hash table is affected by collision and overflow. The hash table collision and overflow can be handled by searching the hash table in some systematic fashion for a bucket that is not full. In open addressing, quadratic and random probing are well-known probe sequence algorithms for collision and overflow resolution. Key density, loading density, loading factor, collisions, overflows, keys clustering, space complexity, and time complexity are the main factors that highly affect the two algorithms during hash table systematic probing. Therefore, this project is conducted to compare the quadratic probing and random probing challenge performance in terms of the key density, loading density, loading factor, overflows, collisions, keys clustering, space complexity, time complexity using step count, the order of magnitude, the worst case, the average case, and the best case. Comparing both algorithms was performed by collecting data from an online survey about the English language proficiency of 104 students. The compression result shows that the random probing algorithm has achieved similar performance compared to quadratic probing in terms of key density, loading density, loading factor, space complexity, order of magnitude, worst case, and average and best case. While the quadratic probing algorithm has recorded less time complexity using the step count method compared to the random probing algorithm. On the other hand, the random probing algorithm has recorded fewer overflows, collisions, and key clustering compared to quadratic probing. However, the study has recommended the quadratic probing algorithm for better time complexity performance and the random probing algorithm for better performance resolving overflows, collisions, and key clustering.

Keywords: Hash map, hash table, quadratic probing, random probing, complexity

1. Introduction

Hashing converts a character string into a typically shorter fixed-length value or key representing the original string. Hashing is used in a database to index and retrieve objects because it is easier to locate the object using the shorter hashed key than using the original value to find it [1]. It is used in many algorithms for encryption as well.

The hash table is a data structure that associatively stores data. Data is stored in an array format in a hash table, where every data value has its own unique index value. If we know the index of the desired data, data access becomes very rapid [2]. Thus, regardless of the size of the data, it becomes a data structure in which insertion and search

*Corresponding author: raghd1985@gmail.com operations are very easy. The hash table uses an array as a storage medium and the hash technique to produce an index from which an element should be inserted or located [1].

The hash table uses a hash function in which each key is allocated to a single bucket. Still, most hash table designs have an incomplete hash function which can cause collisions in hash systems. The hash function produces the same index for more than one key [3]. Hash collisions are almost inevitable when a random subset of a large set of possible keys is hashed. As a result, almost all hash table implementations have some collision resolution strategy to handle such events.

One strategy of collision resolution is known as separate chaining. In this method, each bucket is independent and has some kind of list of entries with the same index. The time for hash table operations is the time for the list operation to find the bucket (which is constant) plus the time. The linked list technique is considered the most famous in a separate chaining strategy. One of the most common methods of collision resolution in hash tables is open addressing or closed hashing. A hash collision is resolved by probing or searching for alternate locations in the array (the probe sequence). With this approach, until either the target record is identified or an unused array slot is found, indicating that the table does not have such a key [4]. Linear probing, double hashing, quadratic probing, and random probing are well-known probe sequences. In this project, quadratic probing and random probing techniques will be addressed in depth.

Quadratic probing is an open address method in computer programming for re-solving hash collisions in hash tables. Quadratic probing works until an open slot is found by taking the original hash index and adding successive values of an arbitrary quadratic polynomial [5]. In an open addressing table, quadratic probing can be a more powerful algorithm as it better avoids the clustering issue that can occur with linear probing. However, it is not immune [6]. It also offers strong memory caching since it retains some reference position. Random probing is a theoretical hashing scheme model based on open addressing and is used to prevent clustering by making the sequence of the probe depend on the key with random hashing [7]. The output of a key generates the sequence of the probe-seeded pseudorandom number generator (possibly together with another component of the seed that is the same for each key but different for different tables) [4]. This paper attempts to compare the performance of quadratic probing and random probing algorithms in modeling hashing technique.

2. Literature Review

In this section, we will summarize five previous studies. These studies each include one of the algorithms that the research is discussing, whether a random or quadratic probing algorithm. These studies are as follows:

A novel approach to PDF-based document steganography is described in this paper. By embedding secret information in between-character positions of words inside cover PDF text, secret communication is achieved by Tyagi et al. [8]. A Stegoencoding technique is designed to enhance the embedding ability along with Stegocover PDF file overheads that are sufficiently reduced. The proposed approach constructs multi-level embedding capabilities that can be used from higher embedding ability to typical embedding capacity according to requirements. Secret text encoding and decoding are achieved through quadratic hashing method testing, which creates a direct encoding table that provides better performance in terms of time complexity. Exploratory results show that the proposed technique offers efficient results in improved security and a high payload of concealed data. Experiential verification and analysis with some popular text steganography techniques reveal that the proposed technique performs better.

An end-to-end real-time adaptive protocol for multimedia transmission is described in Chen et al. [9] work. The transmission rate is calculated by a quadratic probing algorithm that can achieve maximum client buffer usage and minimum network bandwidth allocation. It is also coupled with a mechanism of congestion control that, during network congestion, can effectively reduce the rate of packet loss. In both the Local Area Net (LAN) and the Internet world, they examine the efficiency of our quadratic probing algorithm at different levels of congestion. Analysis of results shows that their strategy is more robust in preventing overload and under-flow at various levels of network congestion and responding to changing delays in the network. Comparisons are made with the approach to fixed rate and the rate by an approach to playback requirements. Experimental findings show that the proposed real-time protocol with the quadratic probing algorithm for rate adjustment successfully uses network resources and reduces packet loss ratios.

In Bello et al. [10] work, a distinction is made between the forms in which collisions are resolved, and circumstances are outlined under which one strategy might be superior to others. Regarding the implementation of hash-based containers, there are two key issues: the hash function and the mechanism for collision resolution. The hash function is responsible for the arithmetic operation that transforms a specific key into a specific table address. The system for collision resolution is responsible for dealing with keys with the same address.

A new approach to studying random probe hashing algorithms is introduced in Bollobás et al. [11] work. The probability-generating function in closed form is extracted for the asymptotic cost of insertion through random probing with secondary clustering. For higher-order clustering, it is shown that all the moments of the insertion cost probability distribution occur and are asymptotically equal to the corresponding moments under uniform hashing of the cost distribution. The method in this paper also leads to simple derivations of the predicted insertion cost of secondary and higher-order clustering for random probing.

The theoretical models of hashing systems focused on open addressing, such as double hashing, are random probing and uniform hashing. An asymptotic analysis of random probe hashing with multi-record buckets was given by Ramakrishna [12]. The Poisson approximation to the binomial distribution was the basis of his research. With the mention of the difficulties involved, the problems of obtaining an exact model and analyzing finite random probing hashing were left open. They discuss these open-ended issues in this article. The search efficiency of complete tables of hash is also analyzed.

3. Research Methodology

Research methodology is the specific procedures or techniques used to identify, select, process, and analyze information about the project. The framework of the research methodology followed in this project includes six stages, as illustrated in fig. 1.

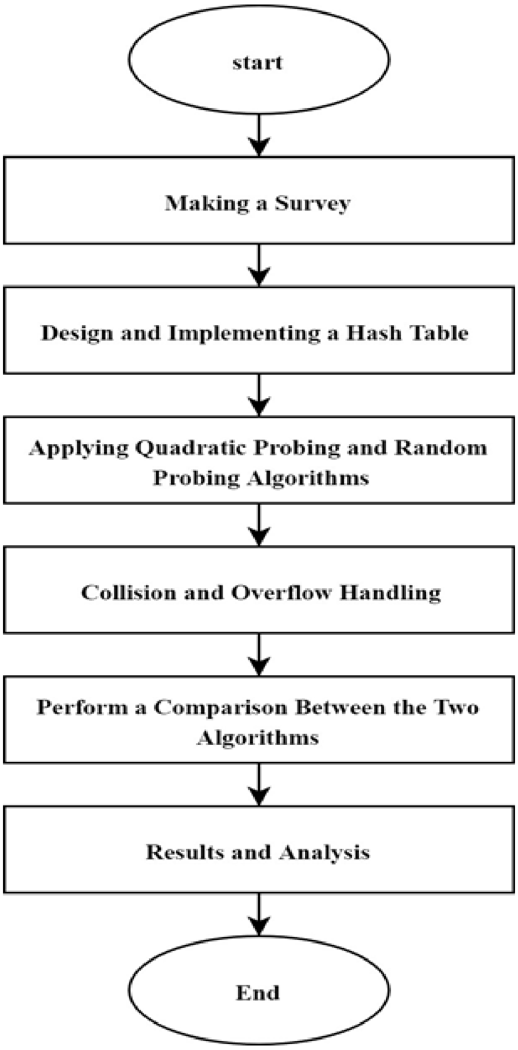

start
Making a Survey
Design and Implementing a Hash Table
Applying Quadratic Probing and Random Probing Algorithms
Collision and Overflow Handling
Perform a Comparison Between the Two Algorithms
Results and Analysis
End

Fig. 1 - Research framework

3.1 The Survey Methodology

The survey was conducted as a questionnaire, a research instrument consisting of a set of questions or other types of prompts that aim to collect information from a respondent. We used a mix of close-ended and open-ended questions in this research questionnaire. One hundred four undergraduate or postgraduate students answered this questionnaire. The questionnaire consists of three introductory questions and nine focused questions on students' English proficiency. Nine questions were closed, and the students had to fill in by clicking on one of the answers. Three of the questions allowed the student to choose one of the available answers or to insert their answer. Only one question requires the students to insert answers. It was about their marks on the formal English test if they had. Table 1 presents the questions of the survey.

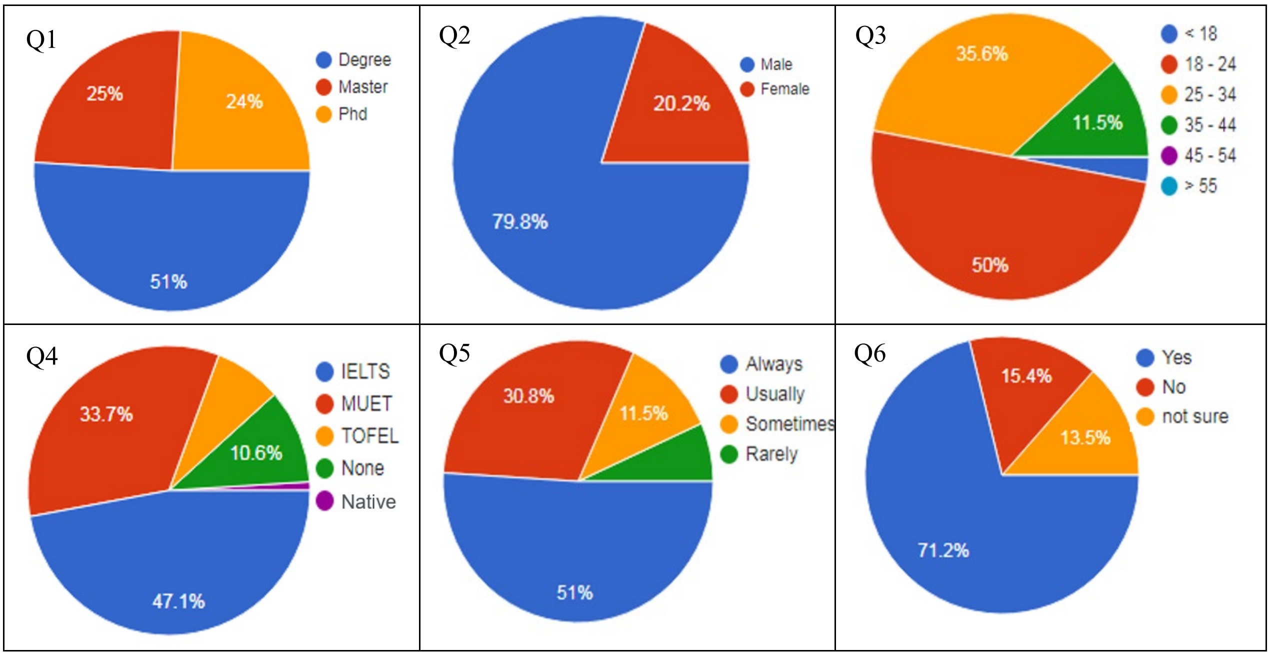
Table 1 - The questionnaires

No.	Question
Q1	Do you currently study?
Q2	What is your gender?
Q3	What is your age group?
Q4	What formal English test did you take?
Q5	Is your English clear and understandable to others?
Q6	Do you like to speak English?
Q7	What do you feel when you speak in front of your lecturer and audience?
Q8	What is your English proficiency out of 9?
Q9	What problems do you face when speaking English with foreign/local students?
Q10	Do you feel motivated to speak English when you see other students speak English fluently?
Q11	What is the best way to improve students' confidence to speak in English?
Q12	What English proficiency level will you reach after completing your study?

Two data collection techniques were used to acquire the collected data. For the closed-ended data, a quantitative data collection approach was used. Compared to the qualitative method, it is simple and can be implemented within a shorter span of time, but the qualitative method was used for open-ended questions. We planned to use the open question to collect various types of data, such as strings, integers, and real numbers, to help insert and search the data in the hash table through the next stages, where quadratic probing and random probing will be executed.

3.2 The Analysis of the Data

Google Forms was arranged to be used to conduct the survey online through social media applications. We planned to post the link to the survey on WhatsApp and Facebook groups, which these groups are allocated for students. The online survey allows the participants to fill in the questions anytime and anywhere using their phones or laptops and provides advantages compared to the paper survey, such as the data being collected and presented in graphs automatically, providing the easiness of analyzing the survey data. The response was quite slow; it took three days to get 104 responses. Fig. 2 provides the results of nine closed-ended and three mixed-ended (open and closed) questions.



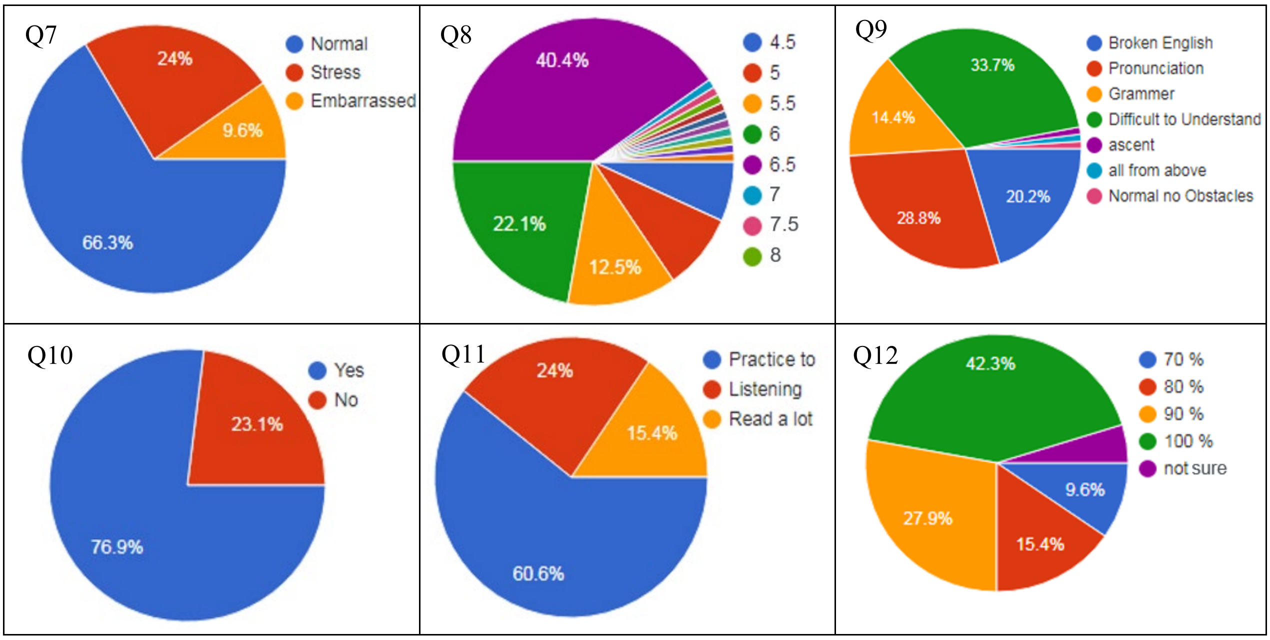


Fig. 2 - Results of the survey

The survey was conducted online to measure the English language proficiency of students between the tenth and twelfth of November 2021. The summary of the survey result provided by the respondents was as follows:

The majority of the respondents in the survey were bachelor's students, as 53 students participated, who account for 51 percent of the total participants, whereas 24 doctoral students and 26 master's students participated, which constituted 24% and 25%, respectively. Twenty-one of the participants were females, which equals about 20%, while 83 were males, representing around 80 %. Fifty-two participants, which comprised half the students, were between the age group of 18 to 24 years old, whereas three students were under 18. The rest of the students were distributed between the two middle-aged groups, where 35.6% were between the 25-34 age groups and 11.5% were between the 35-44 age groups.

Regarding the formal English tests, 49 students took IELTS, and 35 had MUET. Eight students said they took TOEFL, while 12 did not take any tests. 51% of the students said that people always understand them when speaking in English, whereas around 30% said they are usually understood. Also, of 12 students, others sometimes understood their English, which equals around 11%. On the other hand, seven people answered that people rarely understand them when speaking English. Seventy-four respondents said they like to speak English, whereas 14 students were unsure. On the contrary, 16 of the participant prefer to speak in something other than English.

Around 66% of the students speak in English with their lecturers or in front of an audience, whereas the rest feel stressed or embarrassed. 40.4% of students rated themselves 6.5 out of 9 in English language proficiency, whereas 21.1% gave themselves 6. The rest of the students have chosen marks above 6.5 or under 6.

Regarding the problems that the students face when they speak in English with foreign/local students, 35 students chose a difficult-to-understand speaker, and 30 students faced problems with pronunciation. Furthermore, 21 chose broken English, 14 students chose grammar, and the rest chose different reasons. Eighty students said they feel motivated when they see others speaking in English, while 24 were not. Most students, which equals 60.6%, said the best way to improve confidence when someone speaks English is speaking a lot, whereas 24% chose to listen and 15.4% read, respectively. In terms of the proficiency in the English language that the students dream of reaching, 44 students chose 100%, and 29 chose 90%. In addition, 16 students dreamt of reaching 80%, and 10 chose 70%. However, five students were not interested.

3.3 Quadratic Probing Algorithm

Quadratic probing is an open-addressing scheme where we look for an i2'th slot in the i'th iteration if the given hash value x collides in the hash table [13]. The equation can present quadratic probing:

$$(\mathrm{Hash}(x) + i^2)\ \%\ S \tag{1}$$

Let hash (x) be the slot index computed using the hash function. We try if the slot hash (x) % S is full.

$$(\mathrm{Hash}\ (x) + 1*1)\ \%\ S \tag{2}$$

If (hash(x) + 1*1) % S is also full, then we try

$$(\text{Hash }(x) + 2*2)\ \%\ S. \quad (3)$$

If (hash (x) + 2*2) % S is also full, then we try

$$(\text{Hash}(x) + 3*3)\ \%\ S. \quad (4)$$

This process is repeated for all the values of i until an empty slot is found. For example, let us consider a simple hash function as "key mod 7" and a sequence of keys as 50, 700, 76, 85, 92, 73, and 101. This work will design a hash table to insert data from 104 students. The data have been collected through an online survey, which is about their English proficiency. The size of the hash table will be 135 buckets. The hash function consists of the sum of asc11 codes of the student's name as a key mod the table size to assign each key to a unique bucket, which will help to distribute the keys throughout the hash table. Then the quadratic probing algorithm will be used for resolving hash collisions and overflows in hash tables. Quadratic probing operates by taking the original hash index and adding successive values of an arbitrary quadratic polynomial until an open slot is found.

3.4 Random Probing Algorithm

Random probing is a theoretical model of hashing scheme used to avoid clustering by making the probe sequence depend on the key with random hashing. The probe sequence is generated by the output of a pseudorandom number generator seeded by the key [14]. Random probing uses a pseudorandom function to "jump around" in the hash table. The random probe sequence is:

$$h(k,i) = (h[k] + d(i)) \bmod m \quad (5)$$

Where d(0), ..., d(m-1) is a permutation of 0, ..., m - 1. Here, d(i) is an integer, but it is generated uniquely by a recursive function for each i. The recursive definition of d is

$d(0) = 0$

$$d(i+1) = [a*d(i) + 1] \bmod m \quad (6)$$

Where a is a carefully picked integer, the way a is chosen is to ensure each number [0,..., m-1] eventually appears as d(i).

The hash function consists of the sum of asc11 codes of the student's name as a key mod the table size to assign each key to a unique bucket, which will help to distribute the keys throughout the hash table. Then the random probing algorithm will be used to resolve hash collisions in hash tables.

3.5 Collisions and Overflows Handling

Since the collision is part of any hashing technique, our two techniques, quadratic probing, and random probing, are designed to handle this collision, whether randomly or quadratically. In terms of overflow, it can be handled by using the mod. Using mod as the table's size can help bring the overflow hash value to our range of table size easily. Since, in our algorithms, there is no increment in the hash table size, we can do it by setting some loading factor. So, as soon as this limit exceeds the setting load factor, the hash table size will double. This way, we can handle more keys if the hash table is about to get full.

We set the load factor to 0.77, which is under 1, to ensure there will not be overflow, and every key value will find an empty bucket. 0.77 load factor is very near the recommended value, 0.75. This will help insert, delete, and retrieve the data at a suitable time and reduce collisions. In a random probing algorithm, the "a" integer, as shown in Equation 6, must be picked carefully and must be a prime number to generate a permutation number of 0 to m-1. That increases the efficiency of random probing algorithms for handling collision and overflow. The hash function needs to be chosen carefully to distribute the keys throughout the hash table, reducing the time complexity of both algorithms and the collisions and the keys clustering in quadratic probing.

3.6 Evaluation Metrics

The evaluation metrics used in this research contains twelve criteria. They are key density, loading density, loading factor, overflow, collision, key clustering, space complexity, time complexity, order of magnitude, worst case analysis, average case analysis and best case analysis. They have been selected for their ability to assess algorithms. They are well defined in the algorithms assessment related studies of the literature.

4. Evaluation and Results

After the survey had been conducted, the data was collected and presented in the graphs, as shown in Fig. 2. The survey result will be summarized, and the quadratic probing and random probing algorithms have been implemented, tested, and evaluated in this section. The compression result shows that the random probing algorithm has achieved similar performance compared to quadratic probing in terms of key density, loading density, Loading factor, space complexity, order of magnitude, worst case, and average and best case. While the quadratic probing algorithm has recorded less time complexity using the step count method compared to the random probing algorithm. On the other hand, the random probing algorithm has recorded fewer overflows, collisions, and key clustering compared to quadratic probing. However, the study has recommended the quadratic probing algorithm for better time complexity performance and the random probing algorithm for better performance resolving overflows, collisions, and key clustering. The overall comparison between quadratic and random probing functions is shown in Table 2.

Table 2 - Comparison between random and quadratic functions

Quadratic Probing	Comparison	Random Probing
104/135 = 0.77	Key Density	104/135 = 0.77
104/(1 * 135) = 0.77	Loading Density	104/(1 * 135) = 0.77
104/135 = 0.77	Loading Factor	104/135 = 0.77
116 Times	Overflow	53 Times
116 Times	Collision	53 Times
Medium Impact	Key clustering	Low Impact
O(n)	Space complexity:	O(n)
21n + 5	Time Complexity	25n + 6
Assume that, 21n + 5< 21n + 5n when n>1 21n + 5< 26n, when n>1 Then O(n), C=26 and K=1	Order of magnitude using the Big-Oh Notation	Assume that, 25n + 6< 25n + 6n when n>1 25n + 6< 31n, when n>1 Then O(n), c=31and k=1
O(n)	Worst case	O(n)
O(n)	Average case	O(n)
O(1)	Best case	O(1)

5. Conclusion

The project was a comparison study between quadratic probing and random probing algorithms. The two open addressing techniques are used for collision, and overflow resolution when two keys have the same index value or the bucket is full. Two hash tables have been designed to test the performance of each algorithm. The data used in the hash table have been collected online by a questionnaire about the English proficiency of 104 students. The two algorithms have archived the same performance in terms of key density, loading density, loading factor, space complexity, order of magnitude, best case, and verge case. However, the performance of quadratic propping algorithms was better in terms of time complexity using the step count method, which took fewer steps than the random probing algorithm. On the other hand, the random probing algorithm has documented fewer overflows, collisions, and clustering of keys compared to quadratic probing. Nevertheless, the quadratic probing algorithm was recommended in this project over the random probing algorithm in terms of time complexity. In contrast, the random probing algorithm was recommended for better performance resolving overflows, collisions, and key clustering.

References

[1] Aminuddin, A., Saringat, M. Z., Mostafa, S. A., Mustapha, A., Hassan, M. H. (2020). A Case Study on B-Tree Database Indexing Technique. *Journal of Soft Computing and Data Mining*, *1*(1), 27-35.

[2] Maurer, Ward Douglas, and Ted G. Lewis. "Hash table methods." *ACM Computing Surveys (CSUR)* 7.1 (1975): 5-19.

[3] Handschuh, H., & Preneel, B. (2008, August). Key-recovery attacks on universal hash function-based MAC algorithms. In *Annual International Cryptology Conference* (pp. 144-161). Springer, Berlin, Heidelberg.

[4] Jiménez, R. M., & Martínez, C. (2018). On deletions in open addressing hashing. In *2018 Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)* (pp. 23-31). Society for Industrial and Applied Mathematics.

[5] Krishnamurthy, M., Kannan, A., Baskaran, R., & Deepalakshmi, R. (2011). Frequent item set generation using the hashing-quadratic probing technique. *European Journal of Scientific Research*, *50*(4), 523-532.

[6] AbuSalim, S. W., Saringat, M. Z. & Mostafa, S. A., (2020). A Comparative Study of Data Management Systems. *Journal of Soft Computing and Data Mining*, *1*(1), 10-16.

[7] Fichtner, A., & Leeuwen, T. V. (2015). Resolution analysis by random probing. *Journal of Geophysical Research: Solid Earth*, *120*(8), 5549-5573.

[8] Tyagi, S., Dwivedi, R. K., & Saxena, A. K. (2019). A High Capacity PDF Text Steganography Technique Based on Hashing Using Quadratic Probing. *Int. J. Intell. Eng. Syst*, 12(3), 192-202.

[9] Chen, S. C., Shyu, M. L., Gray, I., & Lu, H. (2003, May). An adaptive multimedia transmission protocol for distributed multimedia applications. In *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.* (pp. 537-542). IEEE.

[10] Bello, S. A., Liman, A. M., Gezawa, A. S., Garba, A., & Ado, A. (2014). Comparative Analysis of Linear Probing, Quadratic Probing and Double Hashing Techniques for Resolving Collusion in a Hash Table. *International Journal of Scientific & Engineering Research*, *5*(4).

[11] Bollobás, B., Broder, A. Z., & Simon, I. (1990). The cost distribution of clustering in random probing. *Journal of the ACM (JACM)*, *37*(2), 224-237.

[12] Ramakrishna, M. V. (1989). Analysis of random probing hashing. *Information processing letters*, *31*(2), 83-90.

[13] GeeksforGeeks. (n.d.). Retrieved October 27, 2022, from GeeksforGeeks website: http://www.geeksforgeeks.org

[14] Devroye (n.d.). Retrieved October 27, 2022, from Devroye.org website: http://www.luc.devroye.org