

Bilkent University

CS 202

Fundamental Structures of Computer Science II

FALL 2021

4th Homework

Hashing

Name: Ferhat

Surname: Korkmaz

Section: 01

ID: 21901940

Part -1)

I have implemented my hash table using an array of simple struct Nodes which have two variables, *int item*, *int locationStatus*. *item* is the current content of that hash table slot, whereas, the *locationStatus* indicates the occupation, deletion, or being empty or not as described in the slides. Yet, I could have used negative key values for each of these tags. However, this time, my hash table was not going to support basic negative insertion, deletion, and search operations. Although I did not use or make my main driver function get negative values from my input.txt file in **Part-2** due to the hardness of string processing for that specific case, my hash table itself supports it as I mentioned above.

I declared a couple of private helper functions in order to reduce the recalling same blocks of lines repeatedly such as hash1, hash2, and f.

To resolve conflicts, I have been provided three options such as Linear, Quadratic, and Double probing for open addressing implementation of the hash table.

To avoid infinite loops and inefficient loops for the linear and double operations¹, I stopped the operation when the number of probes reached to the size of hash table, which makes a lot of sense due to every index will be visited at the end and it will come to the first location.

The stopping condition for the quadratic operations is slightly different. This time, whenever the number of probes is equal to half of the size of the hash table, I stopped the operation. This is because if we implement our hash tables primer hashing operation as modular arithmetic with respect to the table size, we generally choose the prime numbers for the efficiency. And If we are dealing with prime number modulas, half of the size of the hash table is highly sufficient.

¹ Operations will be referencing to insertion, removal, and search operations all together.

Part-2)

My hash table's size is 53, which is a prime number, and this is how my experimental input.txt file looks like:

```
S 53      I 71
I 53      I 72
S 53      I 73
R 53      I 74
S 53      I 75
I 53      I 76
I 51      I 78
I 52      I 79
I 59      I 233
I 54      I 277
I 55      I 144
I 56      I 333
I 57      I 130
I 58      I 131
I 2        R 277
I 3        R 144
I 156      I 144
I 166      S 333
I 231      R 333
R 156      S 333
S 231      I 333
           S 333
```

Figure 1: Input.txt file (divided into two since it is a long boi)

For linear probing, my main driver function's output is like the following:

```
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4
g++ -std=c++11 main.o HashTable.o -o hwt
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4$ ./hwt4
53 could not be found after 1 probes
53 has been successfully inserted
53 has been successfully found after 1 probes
53 has been successfully removed
53 could not be found after 2 probes
53 has been successfully inserted
51 has been successfully inserted
52 has been successfully inserted
59 has been successfully inserted
54 has been successfully inserted
55 has been successfully inserted
56 has been successfully inserted
57 has been successfully inserted
58 has been successfully inserted
2 has been successfully inserted
3 has been successfully inserted
156 has been successfully inserted
166 has been successfully inserted
231 has been successfully inserted
156 has been successfully removed
231 has been successfully found after 1 probes
71 has been successfully inserted
72 has been successfully inserted
73 has been successfully inserted
74 has been successfully inserted
75 has been successfully inserted
76 has been successfully inserted
78 has been successfully inserted
79 has been successfully inserted
233 has been successfully inserted
277 has been successfully inserted
144 has been successfully inserted
333 has been successfully inserted
130 has been successfully inserted
131 has been successfully inserted
277 has been successfully removed
144 has been successfully removed
144 has been successfully inserted
333 has been successfully found after 1 probes
333 has been successfully removed
333 could not be found after 2 probes
333 has been successfully inserted
333 has been successfully found after 1 probes
0: 53
1: 54
2: 55
3: 56
4: 57
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4
5: 57
6: 58
7: 59
8: 2
9: 3
10: 166
11:
12:
13:
14:
15: 333
16:
17:
18: 71
19: 231
20: 72
21: 73
22: 74
23: 75
24: 76
25: 78
26: 79
27: 233
28: 130
29: 131
30:
31:
32:
33:
34:
35:
36:
37:
38: 144
39:
40:
41:
42:
43:
44:
45:
46:
47:
48:
49:
50:
51: 51
52: 52
Average number of probes for a successful search: 2
Average number of probes for an unsuccessful search: 4
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4$
```

Figure 2: Output for linear probing

For quadratic probing, my main driver function's output is like the following:

```
fk11@DESKTOP-K3UOO90: /mnt/c/Users/PC/Desktop/HW4$ ./hw4
53 could not be found after 1 probes
53 has been successfully inserted
53 has been successfully found after 1 probes
53 has been successfully removed
53 could not be found after 2 probes
53 has been successfully inserted
51 has been successfully inserted
52 has been successfully inserted
59 has been successfully inserted
54 has been successfully inserted
55 has been successfully inserted
56 has been successfully inserted
57 has been successfully inserted
58 has been successfully inserted
2 has been successfully inserted
3 has been successfully inserted
156 has been successfully inserted
166 has been successfully inserted
231 has been successfully inserted
156 has been successfully removed
231 has been successfully found after 1 probes
71 has been successfully inserted
72 has been successfully inserted
73 has been successfully inserted
74 has been successfully inserted
75 has been successfully inserted
76 has been successfully inserted
78 has been successfully inserted
79 has been successfully inserted
233 has been successfully inserted
277 has been successfully inserted
144 has been successfully inserted
333 has been successfully inserted
130 has been successfully inserted
131 has been successfully inserted
277 has been successfully removed
144 has been successfully removed
144 has been successfully inserted
333 has been successfully found after 1 probes
333 has been successfully removed
333 could not be found after 2 probes
333 has been successfully inserted
333 has been successfully found after 1 probes
0: 53
1: 54
2: 55
3: 56
4: 57
5: 58
6: 59
7: 3
8: 166
9:
10:
11:
12: 2
13:
14:
15: 333
16:
17:
18: 71
19: 231
20: 72
21: 73
22: 74
23: 75
24: 76
25: 78
26: 79
27:
28: 130
29: 131
30: 233
31:
32:
33:
34:
35:
36:
37:
38: 144
39:
40:
41:
42:
43:
44:
45:
46:
47:
48:
49:
50:
51: 51
52: 52
Average number of probes for a successful search: 1
Average number of probes for an unsuccessful search: 2
fk11@DESKTOP-K3UOO90: /mnt/c/Users/PC/Desktop/HW4$
```

Figure 3: Output for quadratic probing

For double probing, my main driver function's output is like the following:

```
Seç fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4$ ./hw4
53 could not be found after 1 probes
53 has been successfully inserted
53 has been successfully found after 1 probes
53 has been successfully removed
53 could not be found after 2 probes
53 has been successfully inserted
51 has been successfully inserted
52 has been successfully inserted
59 has been successfully inserted
54 has been successfully inserted
55 has been successfully inserted
56 has been successfully inserted
57 has been successfully inserted
58 has been successfully inserted
2 has been successfully inserted
3 has been successfully inserted
156 has been successfully inserted
166 has been successfully inserted
231 has been successfully inserted
156 has been successfully removed
231 has been successfully found after 1 probes
71 has been successfully inserted
72 has been successfully inserted
73 has been successfully inserted
74 has been successfully inserted
75 has been successfully inserted
76 has been successfully inserted
78 has been successfully inserted
79 has been successfully inserted
233 has been successfully inserted
277 has been successfully inserted
144 has been successfully inserted
133 has been successfully inserted
138 has been successfully inserted
131 has been successfully inserted
277 has been successfully removed
144 has been successfully removed
144 has been successfully inserted
333 has been successfully found after 1 probes
333 has been successfully removed
333 could not be found after 2 probes
333 has been successfully inserted
333 has been successfully found after 1 probes
0: 53
1: 54
2: 55
3: 56
4: 57
5: 58
6: 59
7: 166
8: 2
9: 3
10:
10:
11:
12:
13:
14:
15: 333
16:
17:
18: 71
19: 231
20: 73
21: 74
22: 75
23: 76
24: 130
25: 78
26: 79
27:
28:
29:
30:
31:
32:
33:
34:
35: 233
36:
37:
38: 144
39:
40:
41:
42:
43:
44:
45:
46: 72
47:
48:
49:
50: 131
51: 51
52: 52
Average number of probes for a successful search: 1
Average number of probes for an unsuccessful search: -1
fk11@DESKTOP-K3U0O90: /mnt/c/Users/PC/Desktop/HW4$
```

Figure 4: Output for double probing

$$\alpha(\text{load factor}) = \frac{\text{current Number of Items}}{\text{table size}}$$

$$\text{my load factor} = \alpha = \frac{26}{53} = 0.49$$

$$\text{Average number of probes of successful search for linear probing} = \frac{1}{2} \left[1 + \frac{1}{1 - \alpha} \right]$$

$$\text{Average number of probes for unsuccessful search for linear probing} = \frac{1}{2} \left[1 + \frac{1}{(1 - \alpha)^2} \right]$$

$$\text{Average number of probes for successful search for quadratic and double probing} = \frac{-\log_e(1 - \alpha)}{\alpha}$$

$$\text{Average number of probes for unsuccessful search for quadratic and double probing} = \frac{1}{1 - \alpha}$$

All of the above equations are taken from the slides.

To make a better look at my data, let me compare my results with the results that are supposed to be by the formulas.

	LINEAR PROBING		QUADRATIC PROBING		DOUBLE PROBING	
	My result	Formula result	My result	Formula result	My result	Formula result
Successful Search	2	1.48	1	1.37	1	1.37
Unsuccessful Search	4	2.42	2	1.96	N/A	1.96

Table 1: My data versus experimental data

Note that Unsuccessful analyze for double probing is N/A since we are asked to return -1 for that case.

Part-3)

Taking a look at Table 1, error rate for the linear probing seems like higher than the other probing methods. That is mainly because linear probing is not that efficient and it causes more conflicts during hashing. Overall, the fact that linear probing's number of probes for successful and unsuccessful search being greater than the other two's number of probes is also provided as it is in the course slides. Also, the error rate is in fact less because we are returning integers for these values and during division a floor operation is performed by the C++ interpreter.