Part 1: Description of Design and Collision Resolution Implementation

Design of the HashTable

- In my design HashTable class has 4 private properties as following;

int tableSize	→ tableSize keeps the size of the hash table
int* hTable	→ in int array hTable, items are kept
int *locType	→ in tint array locType, type of the location is kept Occupied = 1, Empty = 0, Deleted = -1
CollisionStrategy option table	→ option keeps the collision resolution strategy of the

- HashTable class has a constructor, a destructor and 5 public function as following;

HashTable(const int tableSize, const CollisionStrategy option)

- Constructs a HashTable object with given tableSize and collision resolution strategy

~HashTable()

- Destructs the object by deallocating two dynamically allocated int arrays hTable and locType.

bool insert(const int item)

- Inserts the given item in the HashTable and return true if there is a empty location in the table and the item is unique. By using the hash function in a loop, proper index is found and item is inserted in hTable[index]. Thus, location type of the same index changed as occupied. (locType[index] = 1), if empty location is not reached function returns false.

bool remove(const int item)

- Removes the given item from the HashTable and return true if such an item exist in the table. By using the hash function in a loop, index of the given item is found and item is deleted from hTable[index] (hTable[index] = -1). Thus, location type of the same index changed as deleted. (locType[index] = -1), if empty location is reached function returns false.

bool search(const int item, int& numProbes)

- Searchs the given item int the HashTable and return true if such an item exist in the table. By using the hash function in a loop, index of the given item is found, if empty location is reached function returns false.

void display()

- Displays the contents of the table by looping as tableSize.

void analyze(double& numSuccProbes, double& numUnsuccProbes)

- Finds the average number of probes for a successful search and an unsuccessful search of the table in two loop. For the successful search analysis, items in the hashTable are searched and sum of the probes divided by count of existing items. For the unsuccessful search analysis, unexistin items for every index are searched and sum of the probes divided by tableSize.

- HashTable class has 4 protected function for hashing and checking uniqueness as following;

  bool isUnique(const int item)

  - Returns true if given item doesn't exist in the hTable array else returns false.

  int hPrimary(const int key)

  - Hashes the key value as **hash(**key**) =** key **mod** tableSize

  int h1(const int key, const int i)

  - Hashes the key value as **hi**(key) = *hash*(key) + **f**(**i**) mod tableSize According to the collision resolution strategy stored in the option property of the HashTable class, f(i) changes.

  int h2(const int key)

  - Hashes the key value as **hash2**(key) = **reverse**(key)

Collision Resolution Implementation

While probing in insert, remove and search functions there is a risk of infinite loop because probing can cycle through the same sequence of array indices. Stopping condition is designed as checking n positons, n is size of the HashTable. The reason of this design depends on the fact that probe sequence in both 3 strategy repeats after n.

Linear: $(h + 0) \% n = h \% n$

$(h + n) \% n = h \% n$

$(h + (n+1)) \% n = (h + 1) \% n$

$(h + (n+n)) \% n = (h + 2*n) \% n = (h + 1)\%n$

Quadratic: $(h + 0^2) \% n = h \% n$

$(h + n^2) \% n = h \% n$

$(h + 1^2) \% n = (h + 1) \% n$

$(h + (n+1)^2) \% n = (h + n^2 + 2n + 1) \% n = (h + 1)\%n$

Double: $(h + 0*h2) \% n = h \% n$

$(h + n*h2) \% n = h \% n$

$(h + 1*h2) \% n = (h + h2) \% n$

$(h + (n+1)h2) \% n = (h + n*h2 + h2) \% n = (h + h2)\%n$

Part 2: Test of HashTable

Input File

I 10
I 21
I 3
I 56
S 56
I 17
I 39
R 10
R 2
I 3
I 2
I 80
I 7
S 21
S 0
R 81
R 80
I 13
I 111
R 5
I 34
I 25
I 28
I 235
I 1111
I 134
I 912
D 234
I 5
I 75
D 235
S 13
D 13
I 467
I 372

Output 1 with Table Size 29

```
**********QUAD**********
10 inserted
21 inserted
3 inserted
56 inserted
56 found after 1 probes
17 inserted
39 inserted
10 deleted
2 not deleted
Item already exist in table
3 not inserted
2 inserted
80 inserted
7 inserted
21 found after 1 probes
0 not found after 1 probes
81 not deleted
80 deleted
13 inserted
111 inserted
5 not deleted
34 inserted
25 inserted
28 inserted
235 inserted
1111 inserted
134 inserted
912 inserted
234 not found after 3 probes
5 inserted
75 inserted
235 found after 2 probes
13 found after 1 probes
13 found after 1 probes
467 inserted
372 inserted
0:
1:
2:      2
3:      3
4:      235
5:      34
6:      5
7:      7
8:
9:      1111
10:
11:     39
12:     467
13:     13
14:     912
15:
16:
17:     17
18:     134
19:
20:     372
21:     21
22:
23:
24:     111
25:     25
26:     75
27:     56
28:     28
Succ: 1.75
Unsucc: 3.51724
Table Size: 29

**********DOUBLE**********
10 inserted
21 inserted
3 inserted
56 inserted
56 found after 1 probes
17 inserted
39 inserted
10 deleted
2 not deleted
Item already exist in table
3 not inserted
2 inserted
80 inserted
7 inserted
21 found after 1 probes
0 not found after 1 probes
81 not deleted
80 deleted
13 inserted
111 inserted
5 not deleted
34 inserted
25 inserted
28 inserted
235 inserted
1111 inserted
134 inserted
912 inserted
234 not found after 5 probes
5 inserted
75 inserted
235 found after 3 probes
13 found after 1 probes
13 found after 1 probes
467 inserted
372 inserted
0:      912
1:
2:      2
3:      3
4:      467
5:      34
6:
7:      7
8:
9:      1111
10:     5
11:
12:
13:     13
14:
15:     75
16:     39
17:     17
18:     134
19:     372
20:
21:     21
22:
23:     235
24:     111
25:     25
26:
27:     56
28:     28
Succ: 1.6
Unsucc: -1
Table Size: 29

**********LINEAR**********
10 inserted
21 inserted
3 inserted
56 inserted
56 found after 1 probes
17 inserted
39 inserted
10 deleted
2 not deleted
Item already exist in table
3 not inserted
2 inserted
80 inserted
7 inserted
21 found after 1 probes
0 not found after 1 probes
81 not deleted
80 deleted
13 inserted
111 inserted
5 not deleted
34 inserted
25 inserted
28 inserted
235 inserted
1111 inserted
134 inserted
912 inserted
234 not found after 5 probes
5 inserted
75 inserted
235 found after 2 probes
13 found after 1 probes
13 found after 1 probes
467 inserted
372 inserted
0:
1:
2:      2
3:      3
4:      235
5:      34
6:      5
7:      7
8:      467
9:      1111
10:
11:     39
12:
13:     13
14:     912
15:
16:
17:     17
18:     134
19:     75
20:
21:     21
22:
23:
24:     111
25:     25
26:     372
27:     56
28:     28
Succ: 1.65
Unsucc: 3.82759
Table Size: 29
```

Output 2 with Table Size 51

```
**********DOUBLE**********|**********QUAD**********|**********LINEAR**********
10 inserted                |10 inserted              |10 inserted
21 inserted                |21 inserted              |21 inserted
3 inserted                 |3 inserted               |3 inserted
56 inserted                |56 inserted              |56 inserted
56 found after 1 probes    |56 found after 1 probes  |56 found after 1 probes
17 inserted                |17 inserted              |17 inserted
39 inserted                |39 inserted              |39 inserted
10 deleted                 |10 deleted               |10 deleted
2 not deleted              |2 not deleted            |2 not deleted
Item already exist in table|Item already exist in table|Item already exist in table
3 not inserted             |3 not inserted           |3 not inserted
2 inserted                 |2 inserted               |2 inserted
80 inserted                |80 inserted              |80 inserted
7 inserted                 |7 inserted               |7 inserted
21 found after 1 probes    |21 found after 1 probes  |21 found after 1 probes
0 not found after 1 probes |0 not found after 1 probes|0 not found after 1 probes
81 not deleted             |81 not deleted           |81 not deleted
80 deleted                 |80 deleted               |80 deleted
13 inserted                |13 inserted              |13 inserted
111 inserted               |111 inserted             |111 inserted
5 not deleted              |5 not deleted            |5 not deleted
34 inserted                |34 inserted              |34 inserted
25 inserted                |25 inserted              |25 inserted
28 inserted                |28 inserted              |28 inserted
235 inserted               |235 inserted             |235 inserted
1111 inserted              |1111 inserted            |1111 inserted
134 inserted               |134 inserted             |134 inserted
912 inserted               |912 inserted             |912 inserted
234 not found after 1 probes|234 not found after 1 probes|234 not found after 1 probes
5 inserted                 |5 inserted               |5 inserted
75 inserted                |75 inserted              |75 inserted
235 found after 1 probes   |235 found after 1 probes |235 found after 1 probes
13 found after 1 probes    |13 found after 1 probes  |13 found after 1 probes
13 found after 1 probes    |13 found after 1 probes  |13 found after 1 probes
467 inserted               |467 inserted             |467 inserted
372 inserted               |372 inserted             |372 inserted

0:                         |0:                       |0:
1:                         |1:                       |1:
2:      2                  |2:      2                |2:      2
3:      3                  |3:      3                |3:      3
4:                         |4:                       |4:
5:      56                 |5:      56               |5:      56
6:                         |6:      5                |6:      5
7:      7                  |7:      7                |7:      7
8:      467                |8:      467              |8:      467
9:      111                |9:      111              |9:      111
10:     5                  |10:                      |10:
11:                        |11:                      |11:
12:                        |12:                      |12:
13:     13                 |13:     13               |13:     13
14:                        |14:                      |14:
15:     372                |15:     372              |15:     372
16:                        |16:                      |16:
17:     17                 |17:     17               |17:     17
18:                        |18:                      |18:
19:                        |19:                      |19:
20:                        |20:                      |20:
21:     21                 |21:     21               |21:     21
22:                        |22:                      |22:
23:                        |23:                      |23:
24:     75                 |24:     75               |24:     75
25:     25                 |25:     25               |25:     25
26:                        |26:                      |26:
27:                        |27:                      |27:
28:     28                 |28:     28               |28:     28
29:                        |29:                      |29:
30:                        |30:                      |30:
31:     235                |31:     235              |31:     235
32:     134                |32:     134              |32:     134
33:                        |33:                      |33:
34:     34                 |34:     34               |34:     34
35:                        |35:                      |35:
36:                        |36:                      |36:
37:                        |37:                      |37:
38:                        |38:                      |38:
39:     39                 |39:     39               |39:     39
40:     1111               |40:     1111             |40:     1111
41:                        |41:                      |41:
42:                        |42:                      |42:
43:                        |43:                      |43:
44:                        |44:                      |44:
45:     912                |45:     912              |45:     912
46:                        |46:                      |46:
47:                        |47:                      |47:
48:                        |48:                      |48:
49:                        |49:                      |49:
50:                        |50:                      |50:
Succ: 1.05                 |Succ: 1.05               |Succ: 1.05
Unsucc: -1                 |Unsucc: 1.76471          |Unsucc: 1.82353
Table Size: 51             |Table Size: 51           |Table Size: 51
```

Output 3 with Table Size 79

```
*******DOUBLE****    **** QUAD*****        ******LINEAR****

0:                   0:                    0:
1:                   1:                    1:
2:      2            2:      2             2:      2
3:      3            3:      3             3:      3
4:                   4:                    4:
5:      1111         5:      1111          5:      1111
6:                   6:      5             6:      5
7:      7            7:      7             7:      7
8:                   8:                    8:
9:                   9:                    9:
10:     5            10:                   10:
11:                  11:                   11:
12:                  12:                   12:
13:     13           13:     13            13:     13
14:                  14:                   14:
15:                  15:                   15:
16:                  16:                   16:
17:     17           17:     17            17:     17
18:                  18:                   18:
19:                  19:                   19:
20:                  20:                   20:
21:     21           21:     21            21:     21
22:                  22:                   22:
23:                  23:                   23:
24:                  24:                   24:
25:     25           25:     25            25:     25
26:                  26:                   26:
27:                  27:                   27:
28:     28           28:     28            28:     28
29:                  29:                   29:
30:                  30:                   30:
31:                  31:                   31:
32:     111          32:     111           32:     111
33:                  33:                   33:
34:     34           34:     34            34:     34
35:                  35:                   35:
36:                  36:                   36:
37:                  37:                   37:
38:                  38:                   38:
39:     39           39:     39            39:     39
40:                  40:                   40:
41:                  41:                   41:
42:                  42:                   42:
43:     912          43:     912           43:     912
44:                  44:                   44:
45:                  45:                   45:
46:                  46:                   46:
47:                  47:                   47:
48:                  48:                   48:
49:     372          49:                   49:
50:                  50:                   50:
51:                  51:                   51:
52:                  52:                   52:
53:                  53:                   53:
54:                  54:                   54:
55:     134          55:     134           55:     134
56:     56           56:     56            56:     56
57:                  57:     372           57:     372
58:                  58:                   58:
59:                  59:                   59:
60:                  60:                   60:
61:                  61:                   61:
62:                  62:                   62:
63:                  63:                   63:
64:                  64:                   64:
65:                  65:                   65:
66:                  66:                   66:
67:                  67:                   67:
68:                  68:                   68:
69:                  69:                   69:
70:                  70:                   70:
71:                  71:                   71:
72:     467          72:     467           72:     467
73:                  73:                   73:
74:                  74:                   74:
75:     75           75:     75            75:     75
76:                  76:                   76:
77:     235          77:     235           77:     235
78:                  78:                   78:
Succ: 1.15           Succ: 1.1             Succ: 1.1
Unsucc: -1           Unsucc: 1.41772       Unsucc: 1.39241
Table Size: 79       Table Size: 79        Table Size: 79
```

Part 3: Theoretical – Empirical Number of Probe

Formulas for Calculating Average Number of Probes

- **Linear Probing**

$\frac{1}{2}\left[1+\frac{1}{1-\alpha}\right]$ *for a successful search*

$\frac{1}{2}\left[1+\frac{1}{(1-\alpha)^2}\right]$ *for an unsuccessful search*

- **Quadratic Probing & Double Hashing**

$\left[\frac{1}{\alpha}(\log_e \frac{1}{1-\alpha})\right] = \frac{-\log_e(1-\alpha)}{\alpha}$ *for a successful search*

$\frac{1}{1-\alpha}$ *for an unsuccessful search*

TableSize = 29	**LOAD FACTOR α**	**THEORETICAL AVR. NUM. PROBE**		**EMPIRICAL AVR. NUM. PROBE**	
		Successful	Unsuccessful	Successful	Unsuccessful
LINEAR	20/29 = 0.68	2.06	5.38	1.65	3.82
QUADRATIC	20/29 = 0.68	3.56	3.125	1.75	3.51
DOUBLE	20/29 = 0.68	3.56	3.125	1.6	-1

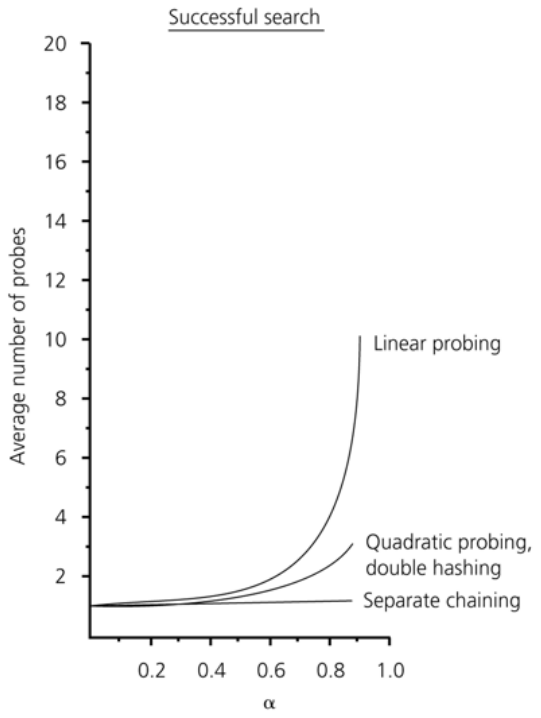
Table1: Output1 Analysis

TableSize = 51	**LOAD FACTOR α**	**THEORETICAL AVR. NUM. PROBE**		**EMPIRICAL AVR. NUM. PROBE**	
		Successful	Unsuccessful	Successful	Unsuccessful
LINEAR	20/51 = 0.39	1.31	1.84	1.05	1.82
QUADRATIC	20/51 = 0.39	1.26	1.63	1.05	1.76
DOUBLE	20/51 = 0.39	1.26	1.63	1.05	-1

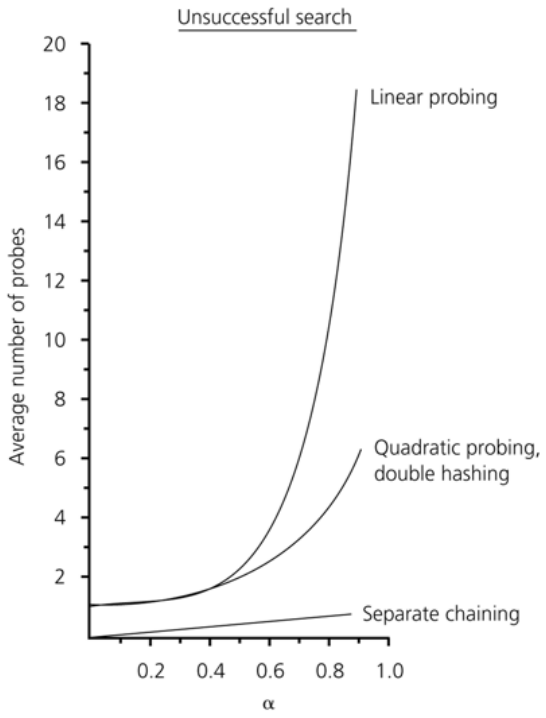
Table2: Output2 Analysis

TableSize = 79	LOAD FACTOR α	THEORETICAL AVR. NUM. PROBE		EMPIRICAL AVR. NUM. PROBE	
		Successful	Unsuccessful	Successful	Unsuccessful
LINEAR	20/79 = 0.25	1.16	1.38	1.1	1.39
QUADRATIC	20/79 = 0.25	1.15	1.33	1.1	1.41
DOUBLE	20/79 = 0.25	1.15	1.33	1.15	-1

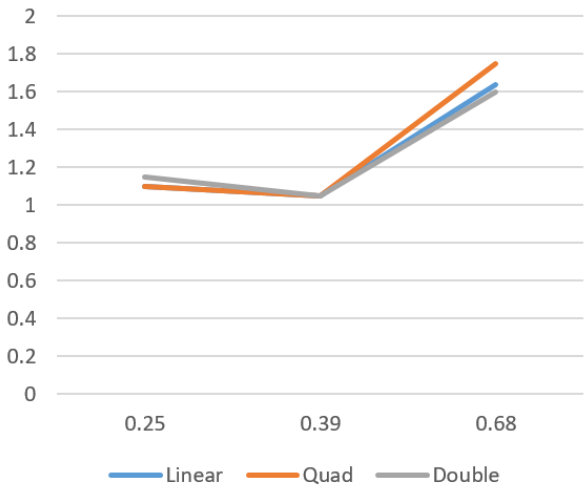
Table3: Output3 Analysis



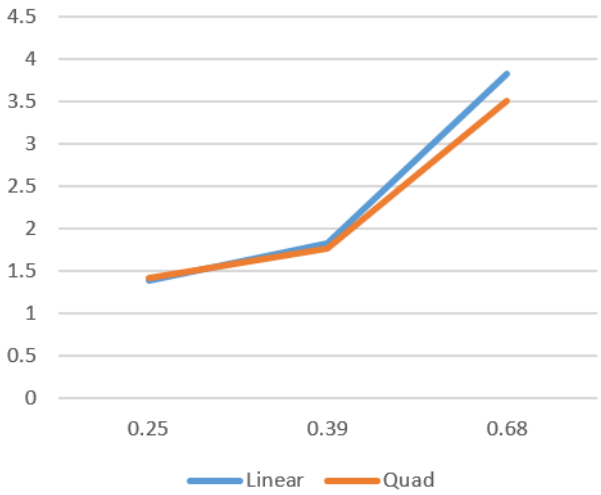
Graph1.a: Successful Search Theoretical



Graph2.a: Unsuccessful Search Theoretical



Graph1.b: Successful Search Empirical



Graph2.b: Unsuccessful Search Empirical

If Graph1.b is observed, it can be seen that while load factor increases it is first decreases than increases rapidly. In theory it should be increasing all the time like in Graph1.a however, the choice of the items and table size can affect the real world result. In the real world example, uniform distribution might not be enough in order to get result close to the theoretical results.

However, when Graph2.a and Graph2.b are observed it can bee seen that empirical result is close to the theory. It's because the probe count in unsuccessful search is less dependent to the uniform distribution due to checking all indexes.