On my honor, I pledge that I have neither recieved nor provided any improper assistance in the completion of this assignment.
- 21900112 김성민

Data Structures:
Hashing & Hash Tables

1. Hashing & Hash Table
2. Collision
3. Rehashing
4. Coding
   - Using list in STL
   - Using unordered_map in STL

Prof. Youngsup Kim, idebtor@gmail.com, CSEE Dept., Grace School Rm204, Handong Global University

Q1. Linear Probing

- Consider a hash table consisting of TableSize = 11, and suppose int keys are hashed into the table using the hash function hash_function(). Suppose that collisions are solved using **linear probing**.

```
int hash_function(int key) {
    int x = (key + 5) * (key + 5);
    x = x / 16;
    x = x + key;
    return x % TableSize;
}
```

- The key listed below are to be inserted, in order given. Show **the home bucket** (to which the key hashes, before any probing), **the probe sequence** (if any) for each key, and **the final hash table** contents.

Key	Home Bucket	Probe Sequence if any
43	0	X
23	6	X
1	3	X
0	1	X
15	7	X
31	2	X
4	9	X
7	5	X
11	5	5→6→7→8
3	7	7→8→9→10

bucket	0	1	2	3	4	5	6	7	8	9	10
key	43	0	31	1	X	7	23	15	11	4	3

Show how you get the probe sequence for the last key 3:

hash_function (3) ⇒ $8 \times 8 = 64$, $64/16 = 4$, $4+3=7$ $7 \% 11 = 7$ ⇒ home bucket

however, there is collision in 7. So, use linear probing.

$bucket_0$ = hash_function(3)%11 = 7 (collision)

$bucket_1$ = (hash_function(3)+1)%11 = 8 (collision)

$bucket_2$ = (hash_function(3)+2)%11 = 9 (collision)

$bucket_3$ = (hash_function(3)+3)%11 = $\boxed{10}$

Q2. Quadratic Probing

Insert sequence: **89, 18, 49, 58, 69**	h(x) = k % 10

	Empty Table	After 89	After 18	After 49	After 58	After 69
0				49 $+1^2$	49	49
1						
2					58 $+2^2$	58
3						69
4						
5						
6						
7						
8			18 $+0^2$	18	18 $+0^2$	18
9		89 $+0^2$	89	89 $+0^2$	89 $+1^2$	89
Unsucessful no. of probes		**0**	**0**	**1**	**2**	**2**

probing sequence

For example, quadratic probing for 58

$h_0(58) = (h(58)+f(0))$ % 10
$= (\quad 8 + 0)$ % 10 = 8 (collision)

$h_1(58) = (h(58)+ 1)$ % 10 = 9 (collision)

$h_2(58) = (h(58)+ 4)$ % 10 = 2

Complete quadratic probing for 69
as explained during the lecture:

$h_0(69) = (h(69)+f(0))$ % 10
$= (9+0)$ % 10 = 9 (collision)

$h_1(69) = (h(69)+ 1)$ % 10 = 0 (collision)

$h_2(69) = (h(69)+ 4)$ % 10 = $\boxed{3}$

Q3. Quadratic Probing

- Consider a hash table consisting of TableSize = 11, and suppose int keys are hashed into the table using the hash function hash_function(). Suppose that collisions are solved using **quadratic probing.**

```
int hash_function(int key) {
    int x = (key + 5) * (key + 5);
    x = x / 16;
    x = x + key;
    return x % TableSize;
}
```

- The key listed below are to be inserted, in order given. Show **the home bucket** (to which the key hashes, before any probing), **the probe sequence** (if any) for each key, and **the final hash table** contents.

Key	Home Bucket	Probe Sequence if any
43	0	X
23	6	X
1	3	X
0	1	X
15	7	X
31	2	X
4	9	X
7	5	X
11	5	5→6→9→3→10
3	7	7→8

bucket	0	1	2	3	4	5	6	7	8	9	10
key	43	0	31	1	X	7	23	15	3	4	11

Show how you get the probe sequence for the last key 3.:

$h_0(3) = (\text{hash\_function}(3) + 0)\ \%\ 11 = 7$ (collision)

$h_1(3) = (\text{hash\_function}(3) + 1)\ \%\ 11 = \boxed{8}$

$(5+1)\ \%\ 11 = 6$ $(5+16)\ \%\ 11 = 10$ $(7+1)\ \%\ 11 = 8$

$(5+4)\ \%\ 11 = 9$

$(5+9)\ \%\ 11 = 3$

Q4. Double Hashing

Insert sequence: **8, 1, 9, 6, 13**

$h(x) = x \% 7$

$h'(x) = R - (x \% R)$
R is prime number less than TableSize

	Empty Table	After 8	After 1	After 9	After 6	After 13
0						
1		8	8	8	8	8
2				9	9	9
3						13
4						
5			1	1	1	1
6					6	6

$h_0(8) = 8 \% 7 = 1$

$h_0(1) = 1 \% 7 = 1$
$h_1(1) = (h(1) + h'(1)) \% 7$ $R=5$
$= (1 + 5 - (1 \% 5)) \% 7 = 5$

$h_0(9) = 9 \% 7 = 2$
$h_0(6) = 6 \% 7 = 6$
$h_0(13) = 13 \% 7 = 6$ (collision)
$h_1(13) = (6 + 5 - (13 \% 5)) \% 7 = 8 \% 7 = 1$ (collision)
$h_2(13) = (6 + 2(5 - (13 \% 5))) \% 7 = 10 \% 7 = 3$

Compute the probe sequence when you insert 13.
Then, its sequence is $6 \to 1 \to 3$.

Q5. Collision – Double Hashing

- Insert keys **43, 25** into the hash table below and find the probe sequence for each:
- Use h(k) = k % 13 with R = 7.

0	1	2	3	4	5	6	7	8	9	10	11	12
26	**None**	**54**	**94**	**17**	**31**	**None**	**None**	**None**	**None**	**None**	**None**	**17**

$h_0(43) = h(43) = 43 \% 13 = 4$ (collision)

$h_1(43) = (4 + 7 - (43 \% 7)) \% 13 = 10 \% 13 = 10$

probe seqence: $4 \to 10$

$h_0(25) = h(25) = 25 \% 13 = 12$ (collision)

$h_1(25) = (12 + 7 - (25 \% 7)) \% 13 = 15 \% 13 = 2$ (collision)

$h_2(25) = (12 + 2(7 - (25 \% 7))) \% 13 = 18 \% 13 = 5$ (collision)

$h_3(25) = (12 + 3(7 - (25 \% 7))) \% 13 = 21 \% 13 = \boxed{8}$

probe sequence: $12 \to 2 \to 5 \to 8$

0	1	2	3	4	5	6	7	8	9	10	11	12
26	x	**54**	**94**	**17**	**31**	x	x	25	x	43	x	**17**

Q6. Rehashing

Rehash the following table into a new hash table below using the hash function:

- **hash(key) = key % 13** and **quadratic probing** to resolve the collisions.
- Show your computation, collision and resolution.
- Compute the load factors before and after rehashing .

$h_0(56) = 56 \% 13 = 4$

$h_0(43) = 43 \% 13 = 4$ (collision)

$h_1(43) = (4+1) \% 13 = 5$

$h_0(30) = 30 \% 13 = 4$ (collision)

$h_1(30) = (4+1) \% 13 = 5$ (collision)

$h_2(30) = (4+4) \% 13 = 8$

$h_0(26) = 26 \% 13 = 0$

$h_0(13) = 13 \% 13 = 0$ (collision)

$h_1(13) = (0+1) \% 13 = 1$

0	1	2	3	4	5	6
56	43	30	None	None	26	13

$\lambda_{before} = \frac{5}{7}$

0	1	2	3	4	5	6	7	8	9	10	11	12
26	13	x	x	56	43	x	x	30	x	x	x	x

$\lambda_{after} = \frac{5}{13}$

Data Structures:
Hashing & Hash Tables

1. Hashing & Hash Table
2. Collision
3. Rehashing
4. Coding
   - Using list in STL
   - Using unordered_map in STL

Prof. Youngsup Kim, idebtor@gmail.com, CSEE Dept., Grace School Rm204, Handong Global University