1. Dealing with Collisions

(a)

- **Q:** For a hash table of size m with n keys, if $n = 2m$ and the keys are not evenly distributed and separate chaining is used to resolve collisions, what is the worst case runtime complexity of a search for a specific key using big O notation?
- **A:** $O(n)$，当所有键的hash值模 m 全部一样的时候，会形成一条长为 n 的链，在上面查询最坏是 $O(n)$ 的。
- **Q:** For a hash table of size m with n keys, if $n = 2m$ and the keys are evenly distributed and separate chaining is used to resolve collisions, what is the worst-case runtime complexity of a search for a specific key using big O notation?
- **A:** $O(1)$，因为键值是均匀分布的，每次查询最多只会在链上走 2 次。

(b)

- **Q:** Assume that we hash a set of integer keys into a hash table of capacity $m = 13$ using a hash function $h(k) = k$.

  Show how the set of keys below will be stored in the hash table by drawing the ***final*** state of each chain of the table after all of the keys are inserted, one by one, in the order shown. If a chain is empty, indicate NULL where appropriate.

  `54, 23, 67, 88, 39, 75, 49, 5`

- **A:**

$$
\begin{gathered}
54 \ mod \ 13 = 2 \\
23 \ mod \ 13 = 10 \\
67 \ mod \ 13 = 2 \\
88 \ mod \ 13 = 10 \\
39 \ mod \ 13 = 0 \\
75 \ mod \ 13 = 10 \\
49 \ mod \ 13 = 10 \\
5 \ mod \ 13 = 5
\end{gathered}
$$

```
    ----
0 | 39 |
    ----
1 |NULL|
    ---- ----
2 | 54 | 67 |
    ---- ----
3 |NULL|
    ----
4 |NULL|
    ----
5 |  5 |
    ----
6 |NULL|
    ----
7 |NULL|
    ----
```

```
 8 |NULL|
    ----
 9 |NULL|
    ---- ---- ---- ----
10 | 23 | 88 | 75 | 49 |
    ---- ---- ---- ----
11 |NULL|
    ----
12 |NULL|
    ----
```

(c)

- **Q:** Show where the sequence of keys shown below are stored in the hash table if they are inserted one by one, in the order shown, with $h(k) = k$ and $m = 13$, using linear probing to resolve collisions.

  `54, 23, 67, 88, 39, 75, 49, 5`

- **A:**

```
   0    1    2    3    4    5    6    7    8    9   10   11   12
 ----------------------------------------------------------------
| 39 | 49 | 54 | 67 |    |  5 |    |    |    |    | 23 | 88 | 75 |
 ----------------------------------------------------------------
```

(d)

- **Q:** Show where the sequence of keys shown below are stored in the hash table if they are inserted one by one, in the order shown, with $h(k) = k$ and $m = 13$, using quadratic probing to resolve collisions.

  `54, 23, 67, 88, 39, 75, 49, 5`

- **A:**

```
   0    1    2    3    4    5    6    7    8    9   10   11   12
 ----------------------------------------------------------------
| 39 | 75 | 54 | 67 |    |  5 | 49 |    |    |    | 23 | 88 |    |
 ----------------------------------------------------------------
```

(e)

- **Q:** Quadratic probing suffers from one problem that linear probing does not. In particular, given a non-full hashtable, insertions with linear probing will always succeed, while insertions with quadratic probing may or may not succeed (i.e. they may never find an open spot to insert). Using $h(k) = k$ as your hash function and $m = 6$ as your table capacity, give an example of a non-full hashtable and a key that cannot be successfully inserted using quadratic probing.
- **A:** (HINT: Start by inserting the keys 36, 78, 12, 90.)

$$
\begin{aligned}
0^2 \; mod \; 6 = 0\\
1^2 \; mod \; 6 = 1\\
2^2 \; mod \; 6 = 4\\
3^2 \; mod \; 6 = 3\\
4^2 \; mod \; 6 = 4\\
5^2 \; mod \; 6 = 1
\end{aligned}
$$

$i^2 \; mod \; 6$ 只能取 $0, 1, 3, 4$，在同一位置插入 4 次后无法继续处理冲突。

```
      0     1     2     3     4     5       Key to insert: ___18___
    -----------------------------
    | 36 | 78 |    | 90 | 12 |    |
    -----------------------------
```

2. Strings as Keys

(a)

- **Q:** If 15105 strings were stored in a hash table of size 3021 using separate chaining, what would the load factor of the table be? If the strings above were equally distributed in the hash table, what does the load factor tell you about the chains?
- **A:** $\alpha = n/m = 5$，负载因子是 5。每个下标处都有一条链，每条链长度是 5。

(b)

- **Q:** Using the hash function above with a table size of 3021, give an example of two different strings that would "collide" in the hash table and would be stored in the same chain. Show your work. (Use short strings please!)
- **A:** $31^3 = 29791 > 3021$，两个长度为 3 的串就可能发生碰撞。

  如 `aaa` 和 `deo` 会发生碰撞。