Analysis of Hash Table

For **good hash function**, the output of the statistics should have:

- 1/tablesize of conflict_count, as a good hash function should have minimum conflict.
- low probe_total, it should minimize the chance of probing.
- short probe_max, it should not probe very long.

For **bad hash function**, the output of the statistics should have:

- high conflict_count, it should hash a lot of value to the same key
- high probe_total, it should probe very often.
- long probe_max, it should probe a very long distance

Code used to generate random value to be hashed. Randomly generate strings to be stored in hash table, and do it for 1000 to 10000 items.

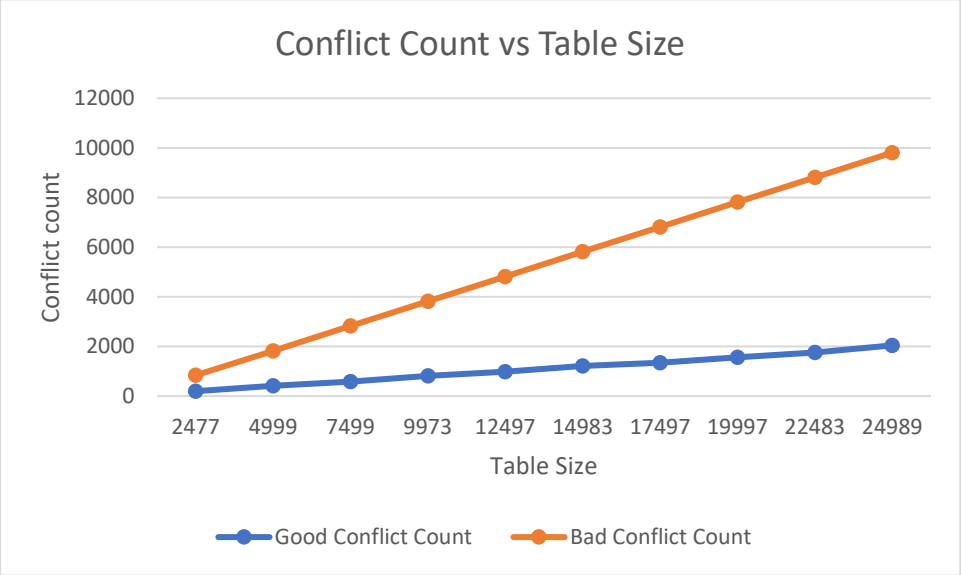
```
import random
import string
import pandas as pd
from hash_table import LinearProbePotionTable
x = []
for i in range(1000, 10001, 1000):
    random.seed(1)
    y = [''.join(random.choices(string.ascii_letters + string.digits, k= 10)) for _ in range(i)]
    good = LinearProbePotionTable(len(y), True)
    bad = LinearProbePotionTable(len(y), False)
    for i in y:
        good[i] = i
    for i in y:
        bad[i] = i
    x.append((good.table_size, good.statistics()[0], good.statistics()[1], good.statistics()[2], bad.statistics()[0], bad.statistics()[1], bad.statistics()[2],))

df = pd.DataFrame(x, columns=['Table size', 'Good Conflict', 'Good Probe Total', 'Good Probe max', 'Bad Conflict', 'Bad Probe Total', 'Bad Probe Max'])
df.to_csv("Data.csv")
```

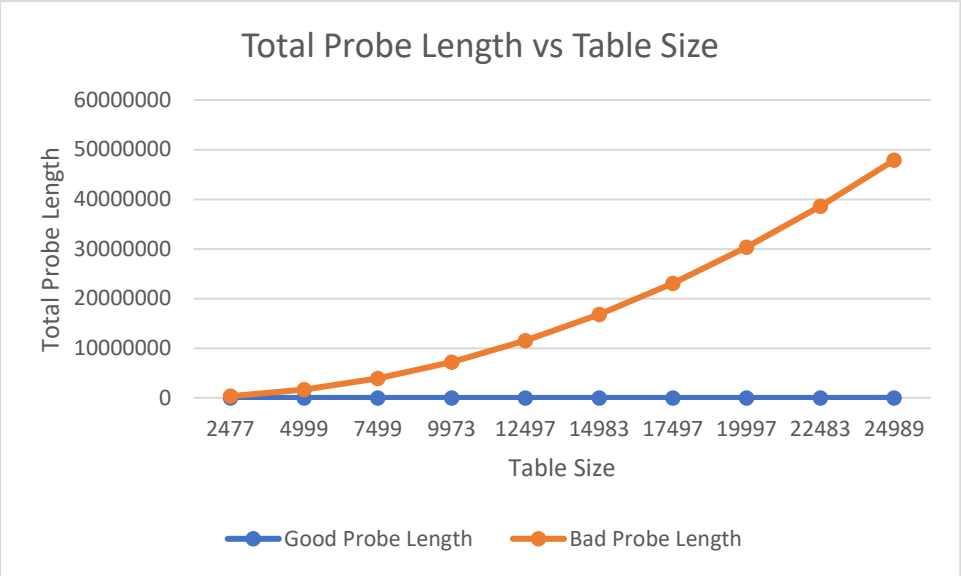
Statistics result from hash table

Table Size	Good Conflict Count	Good Probe Length	Good Probe Max	Bad Conflict Count	Bad Probe Length	Bad Probe Max
2477	199	327	11	840	340505	934
4999	418	702	12	1827	1629654	1906
7499	579	1012	18	2825	3928185	2955
9973	817	1386	14	3821	7218377	3943
12497	981	1662	10	4818	11503173	4924
14983	1214	2039	12	5818	16799900	5921
17497	1349	2350	12	6815	23063355	6954
19997	1563	2512	13	7815	30337154	7898
22483	1757	2874	19	8812	38594300	8952
24989	2045	3458	14	9812	47880174	9902

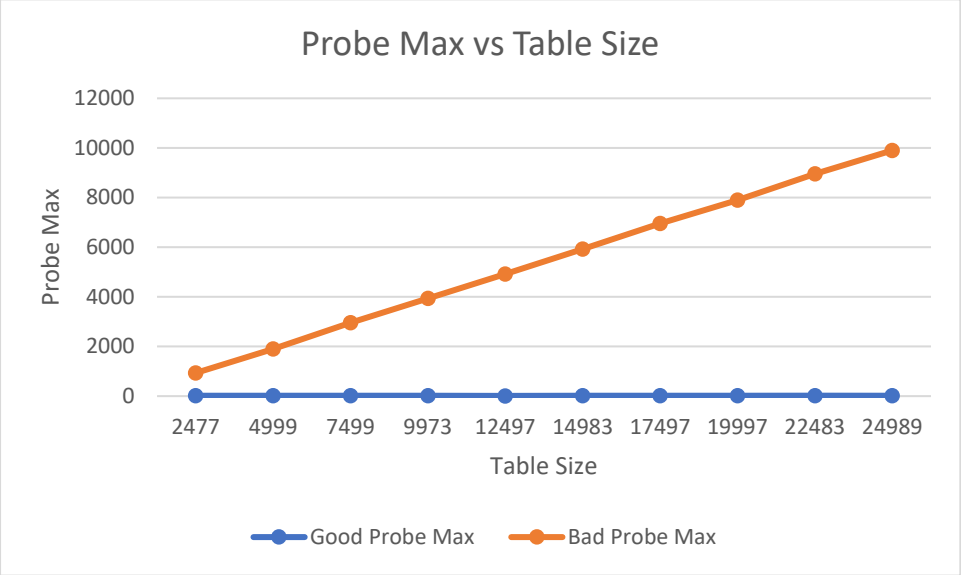
As we can see from the table above and chart below, the conflict count for most of the good hash isless than 1/ tablesize, hence the conflict count for good hash is low. For bad hash, the conflict count is about tablesize /3 for every table size, hence it has relatively high conflict count.



By comparing probe total of good hash and bad hash, we can see that probe total of good hash does not grow as much as bad hash, and it is kept low. For bad hash, it grows exponentially and it probed more than ten times of the table, so the probe total of bad hash is very high.



By comparing probe max of good hash and bad hash, we can see that the good hash probe max does not increase with the table size, and is kept low for every table size. Probe max for bad hash increases with the table size and almost it almost probed half to the table which is really high.



According to the data and statistics, we can validate that our prediction about good hash and bad hash are accurate.