# HASH TABLE ANALYSIS
# GROUP T01G01

## ***CODE***

```
def analysis(item_size, good_hash=True):
    random.seed(0)
    hash_table = LinearProbePotionTable(item_size, good_hash)

    range_lst = []
    for i in range(len(names_list)):
        range_lst.append(i)

    for i in range(item_size):
        x = random.choice(range_lst)
        range_lst.remove(x)
        hash_table[names_list[x]] = names_list[x]

    out = (hash_table.statistics())
    return out

txt = \
    “...”  # Shortened since very many lines
names_list = txt.split(",")
```

## ***APPROACH TAKEN***

We have parameters item size to change the number of items from list taken and good_hash which is boolean to set to either good or bad hash. We import random and set the seed to 0 so that the random numbers are always the same. We create the hash table by calling LinearProbePotionTable. We have a temporary list where we append numbers till 1000 which is the number of names in our name list. Then we use another for loop in the range of item_size parameter where we use the random choice method to pick a random number in the range 1000 as it will be used as the index to pick an item from the list. We also remove this number from the list of range 1000 so this number isn't picked again. Next, we hash the name by setting the key from the name list with index x and set data as the chosen name. Lastly, we just return the statistics method which prints the Conflict Count, Probe Total, Probe Max

The testers at the bottom of the file change item range from 10 to 1000 and also check for the same items whats the difference for Good Hash and Bad Hash and print both. The text is a comma-separated paragraph so we need to use txt.split to make 1000 words separate strings in the list.

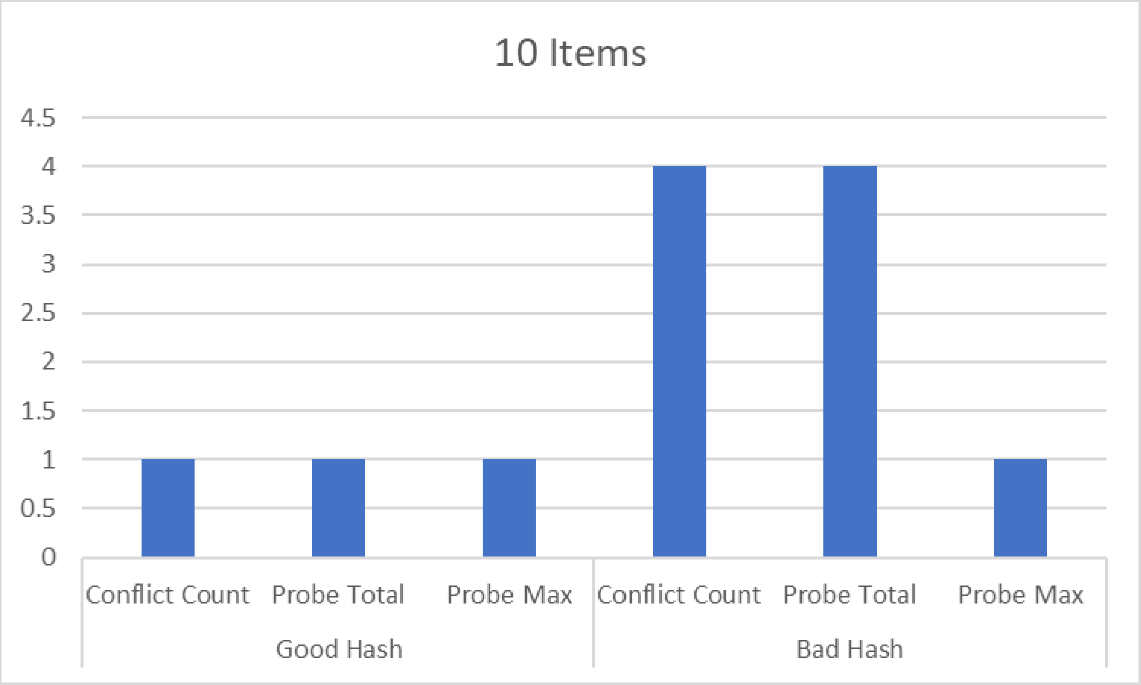
## ***EXPLANATION***

We are tasked with finding out the difference between the output of the Good Hash function and the Bad Hash function so what we will be doing is using the defined function analysis in the file hash_table_analysis.py. Since we have 1000 items and we can find out the difference between Good and Bad hash by changing the number of item_size parameter and evaluating the difference between the statistics returned by good and bad hash. The fake data, in this case, is the 1000 names retrieved from the online source below.

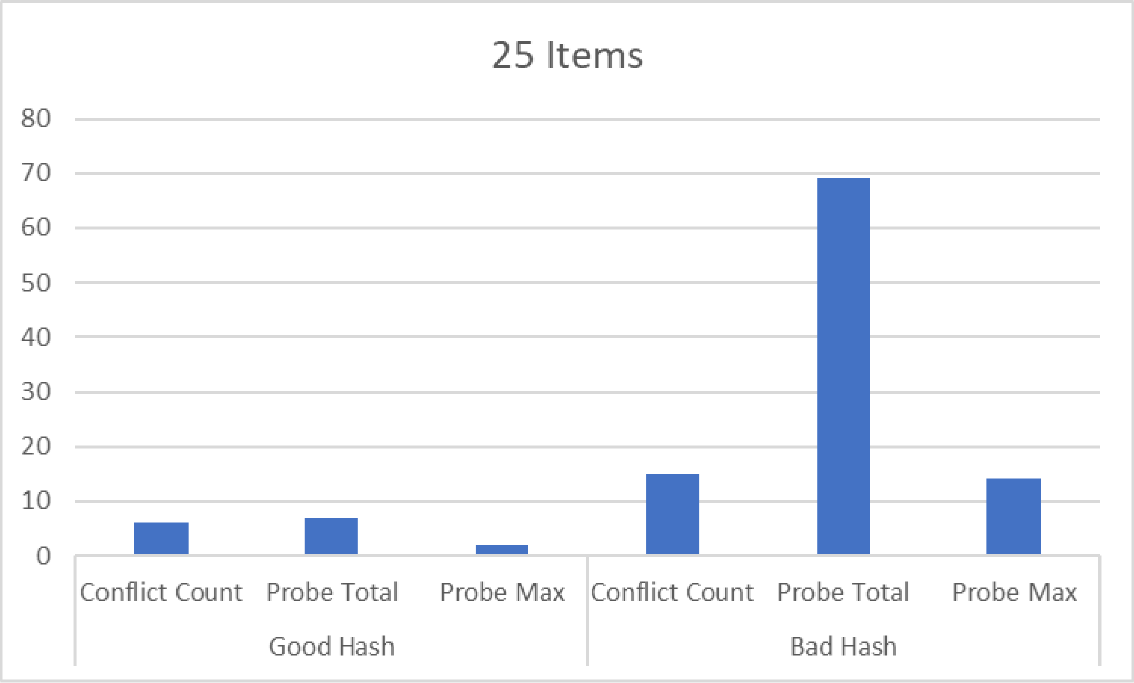
(https://www.usna.edu/Users/cs/roche/courses/s15si335/proj1/files.php%3Ff=names.txt.html)

## ***DATA***

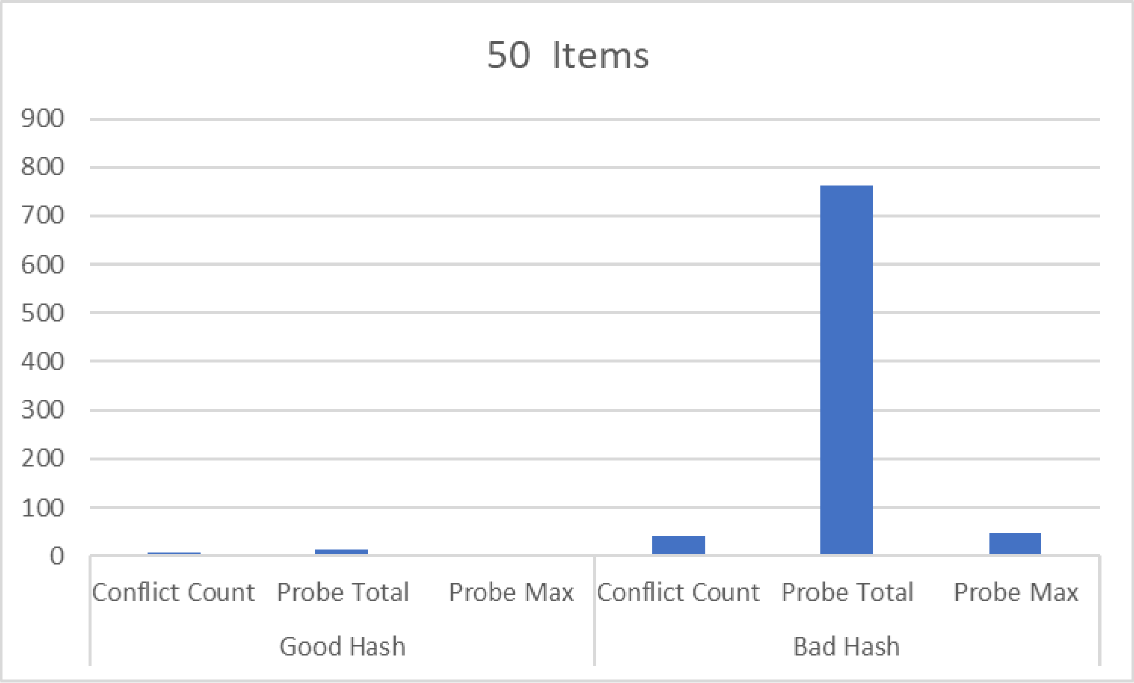
***Note: Statistics are in the order of conflict count, probe total and probe max***



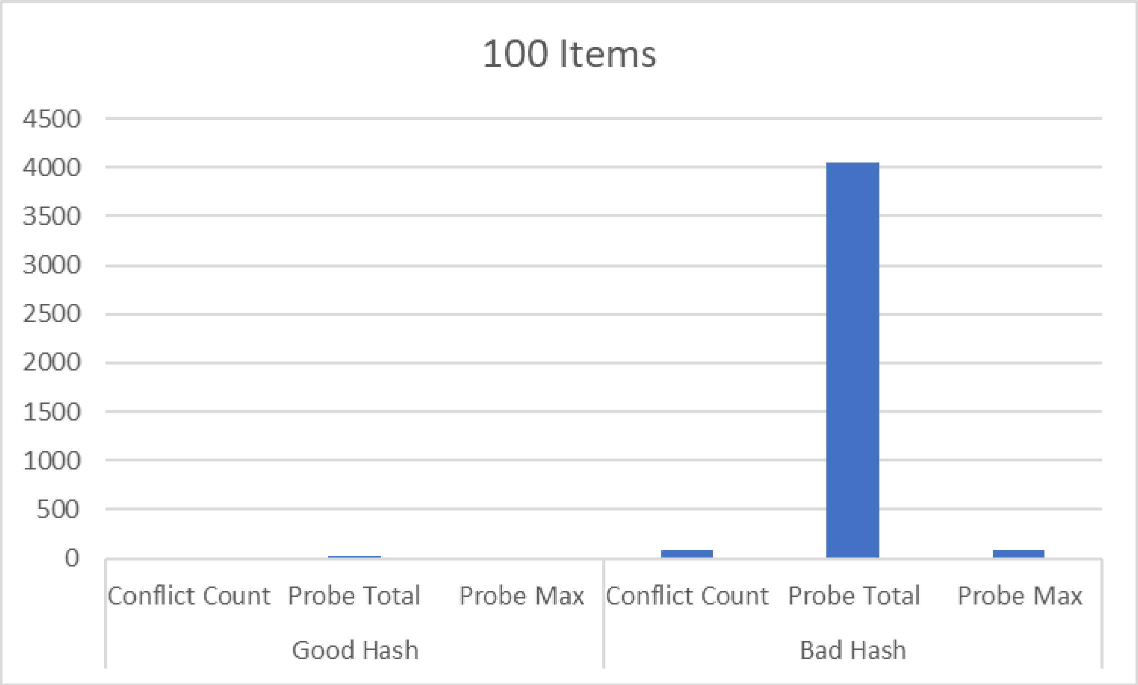
- When we have 10 Items for hashing, the statistics for the good hash is 1,1,1 and for the bad hash, it is 4,4,1. With such a small number of items the difference in statistics is really small and can be considered to be negligible.



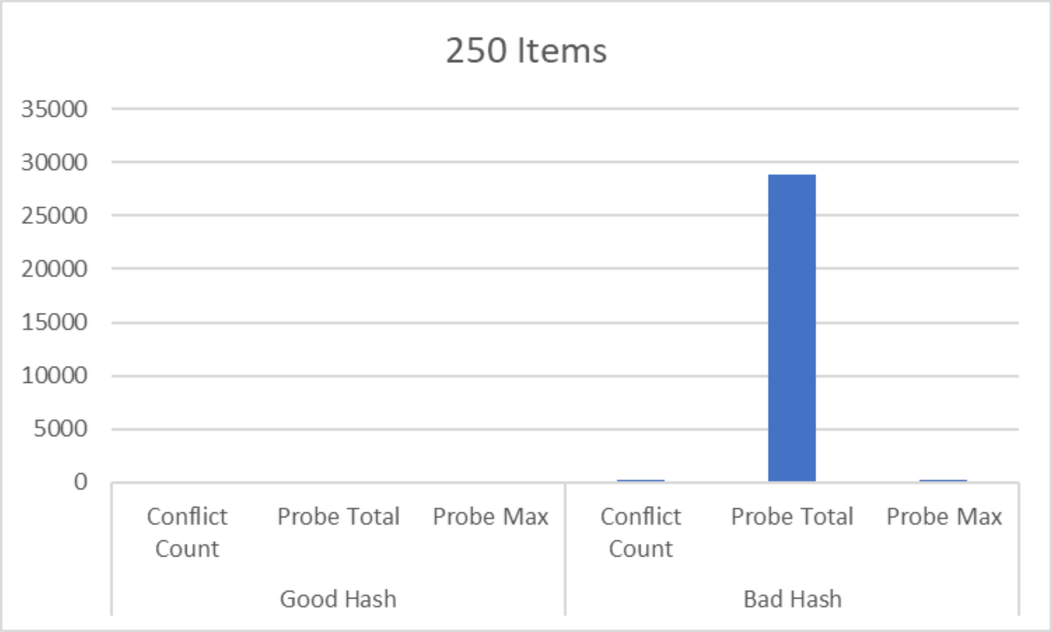
- When we increase items to 25, good and bad are 6,7,2 and 15,69,14 respectively. We can begin to see the difference. While conflict count is still in small difference the probe total of the bad hash is ten times of probe total in the good hash.



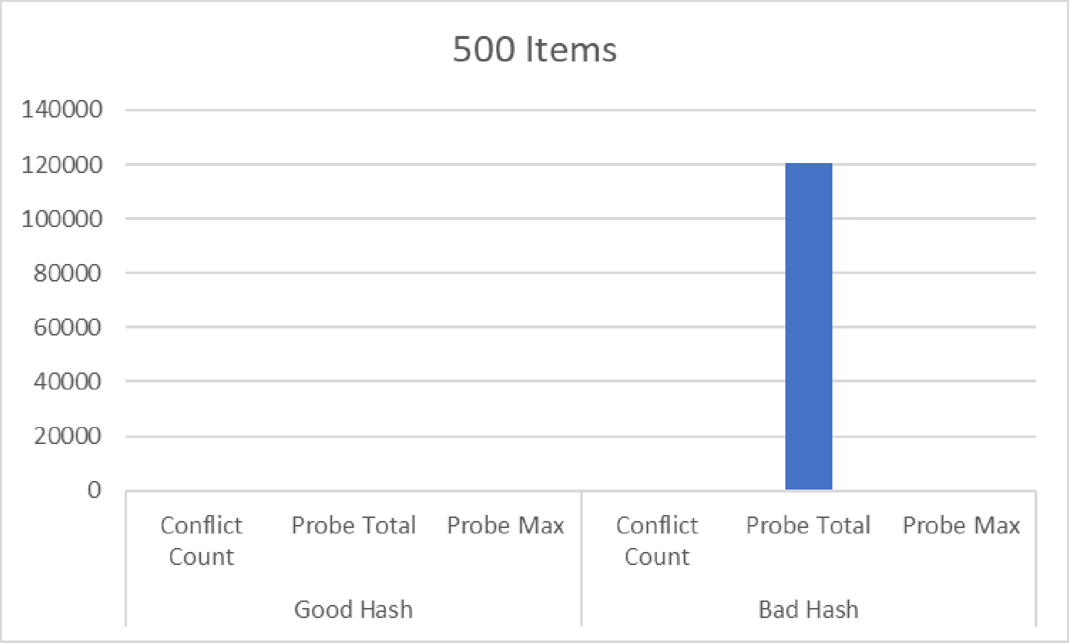
- When the item count is 50, the good and bad hash are 6,12,5 and 40,763,47 respectively. Here the difference between the probe total is really huge for the good hash probe total is only 12 while for the bad hash is 763. The probe max is only quite different from 5 to 47 and conflict also goes from 6 to 40.



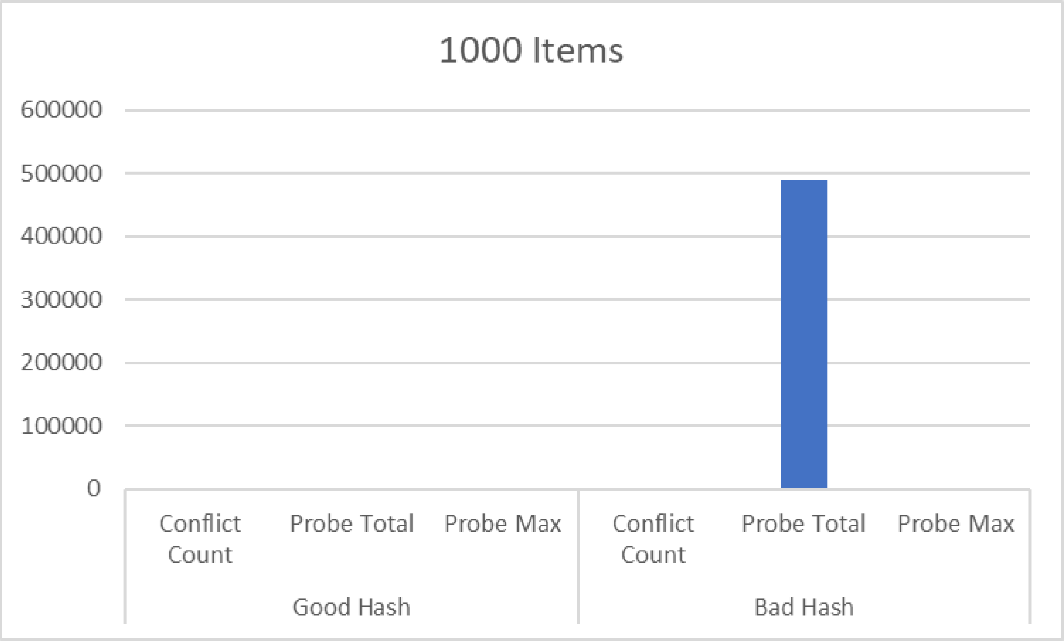
- When the item count is 100, the good and bad hash are 14,21,3 and 90,4045,95 respectively. Here the difference between the probe total is really huge for the good hash probe total is only 21 while for the bad hash is 4045 this is approximately 200 times more. And now the difference is really clear. From the graph, we can't even see conflict count and probe max the graph/



- Good Hash is 40, 70, 5 and Bad Hash is 240, 28915, 243. Now even the ratio between the bad hash and the good hash becomes to be large enough. The difference between all columns is really large. Again only the probe total can be seen on the graph since it's 28915.



- Good Hash is 80, 103, 5 and Bad Hash is 490, 120361, 496. Here the probe total reaches more than 120361 which is a really big number. This is also the first time the output isn't output to the console instantly since it takes the computer a while to process the data.



- Finally for 1000 items, Good Hash is 170, 247, 9 and Bad Hash is 990, 490521, 997. The probe total increased 5 more times from 500 items for bad hash which was already huge, to begin with. The good hash also increases a bit for this part. For 1000 items, even the good hash takes a while to output and the bad hash takes way more time which is understandable seeing the data

In conclusion the good hash is way better than the bad hash which can already be seen when items are around 100 and when item count becomes 1000 the difference is really big shown by data.