Assignment 1 (Report)
Implementation to databases

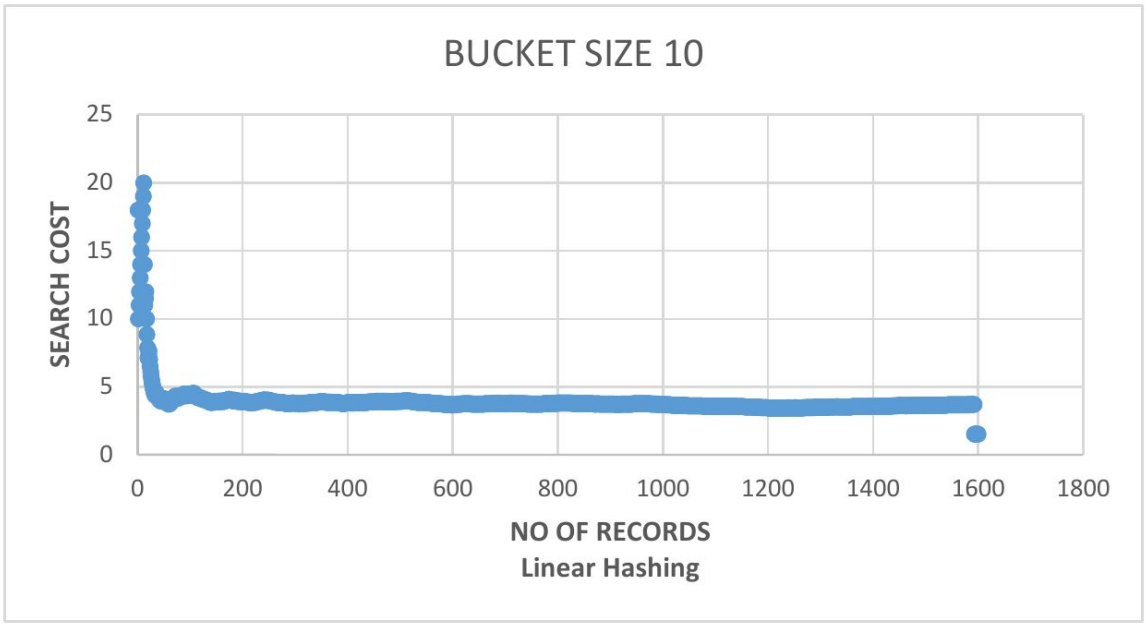
kartik vishwakarma
2017csm1001

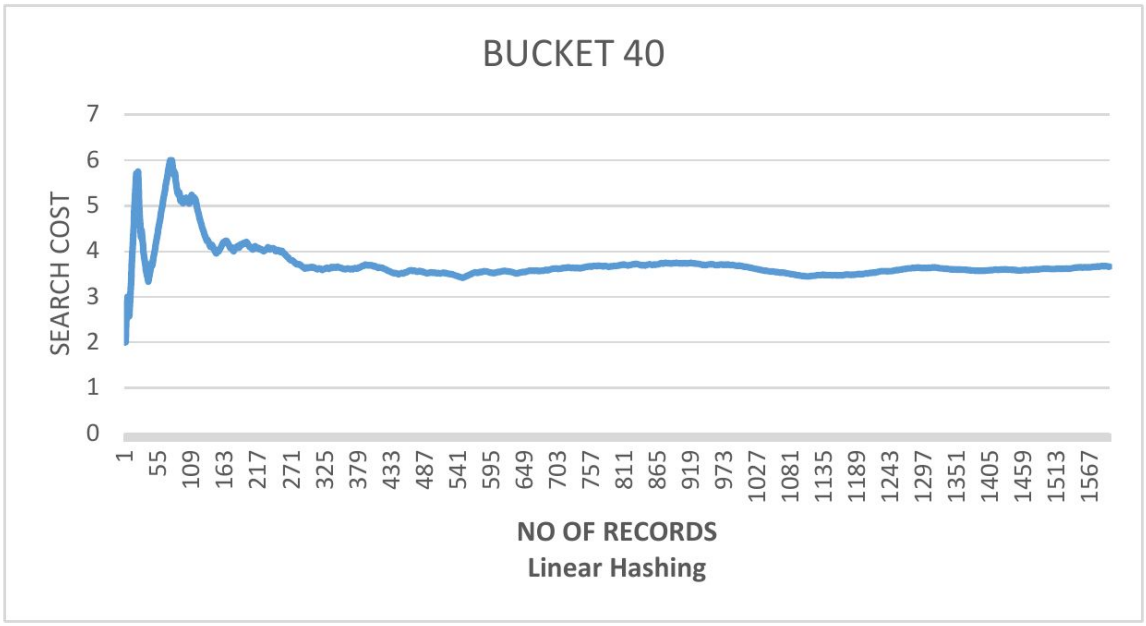
Q1: Implementation of Linear and Extendible Hashing.

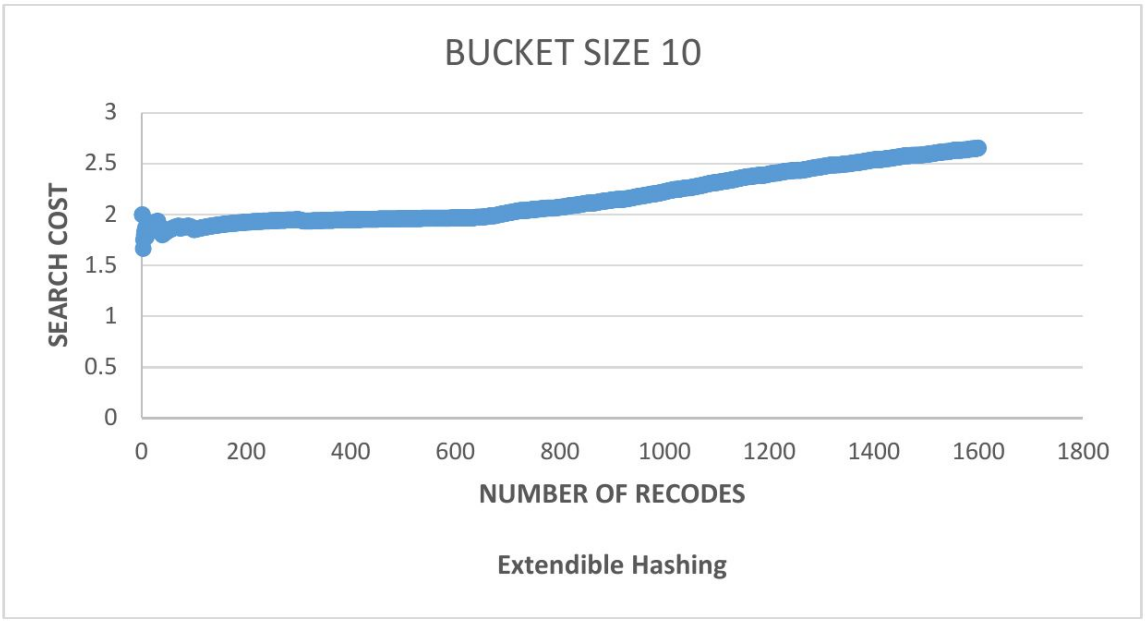
- Both implementation are iterative in nature, only one abstract data type is used.
- A large array of abstract data type are stored in primary memory treating it as secondary memory for simulation.
- C++ is used as language for implementation.
- Dataset are generated randomly using rand() function. Data set contains unique keys.

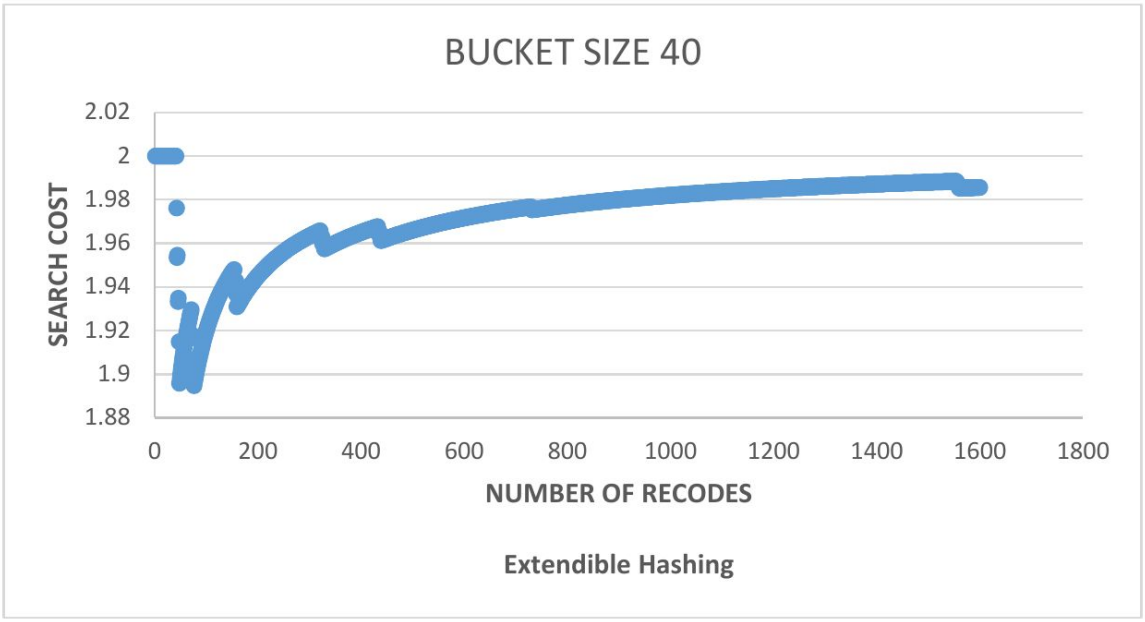
Note: LSB bit used in Extendible hashing instead of MSB for sack of implementation and verification (debugging).

Search cost:


BUCKET SIZE 10
SEARCH COST
25
20
15
10
5
0
0
200
400
600
800
1000
1200
1400
1600
1800
NO OF RECORDS
Linear Hashing


BUCKET 40
SEARCH COST
7
6
5
4
3
2
1
0
1
55
109
163
217
271
325
379
433
487
541
595
649
703
757
811
865
919
973
1027
1081
1135
1189
1243
1297
1351
1405
1459
1513
1567
NO OF RECORDS
Linear Hashing



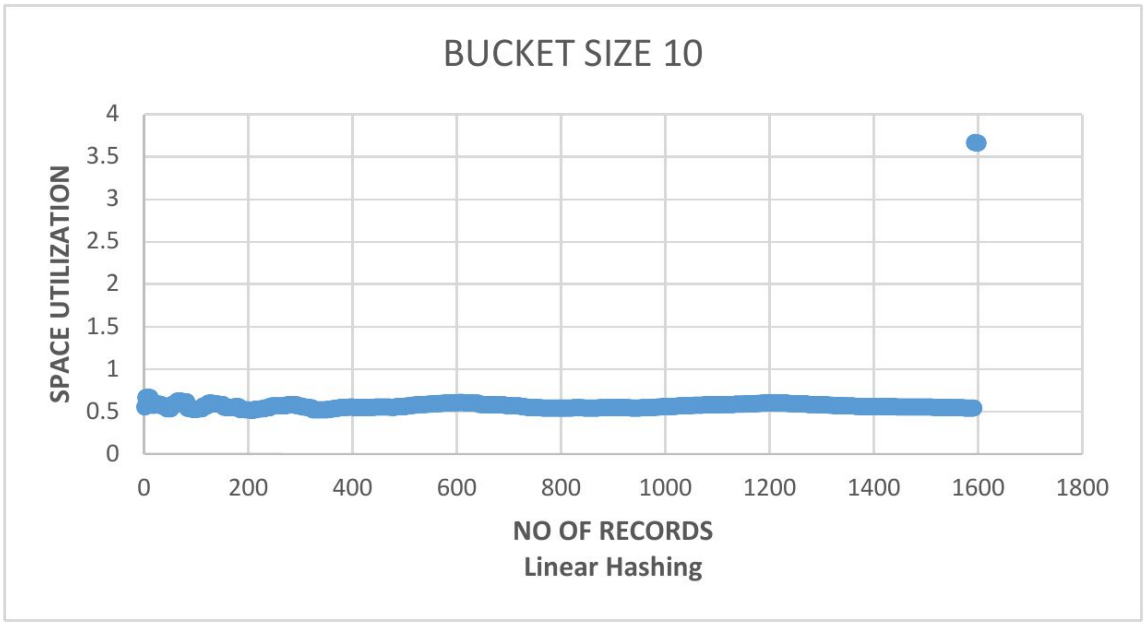


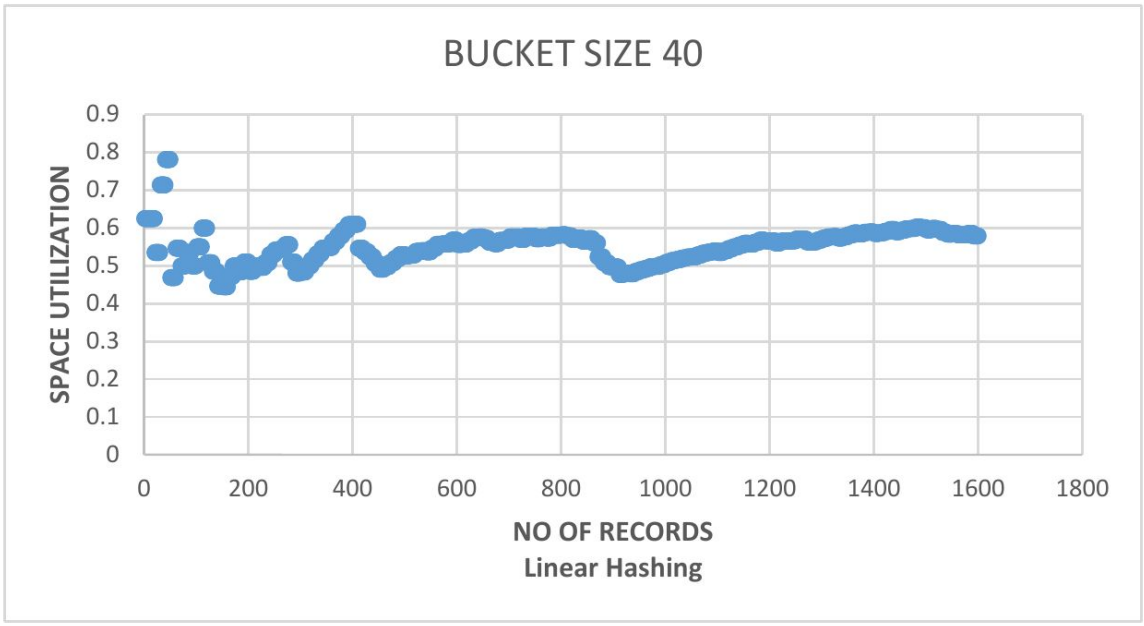
Above plot show searching cost of Linear and extendible hashing using bucket size 10 and 40 respectively.

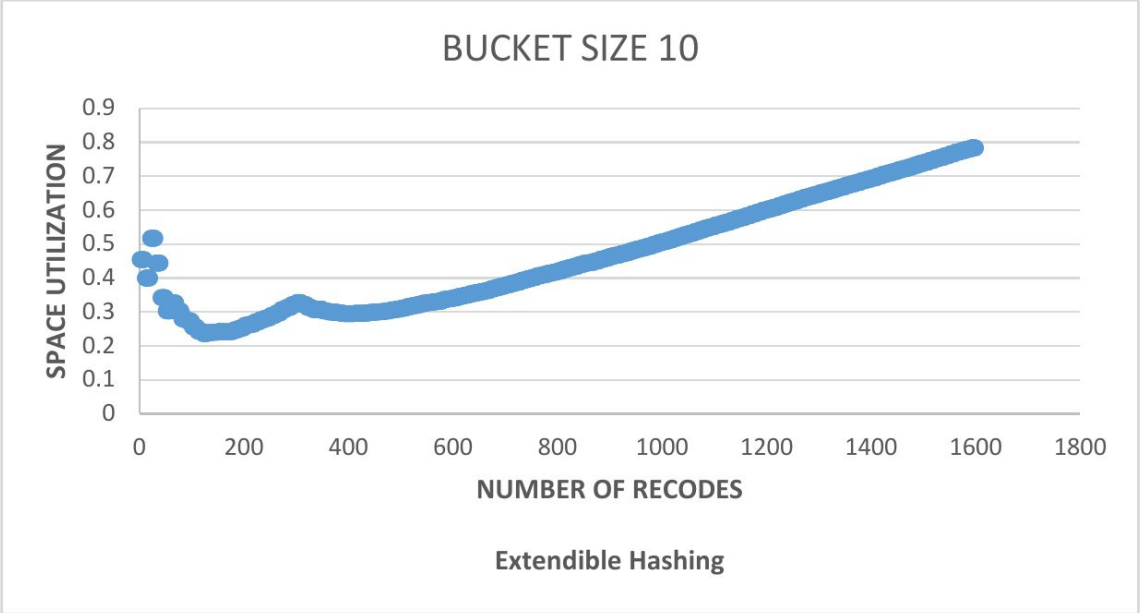
Looking at bucket size Both hashing with bucket size 40 having less search cost than bucket size 10.
Naturally, data with more entry per bucket will reduce search space

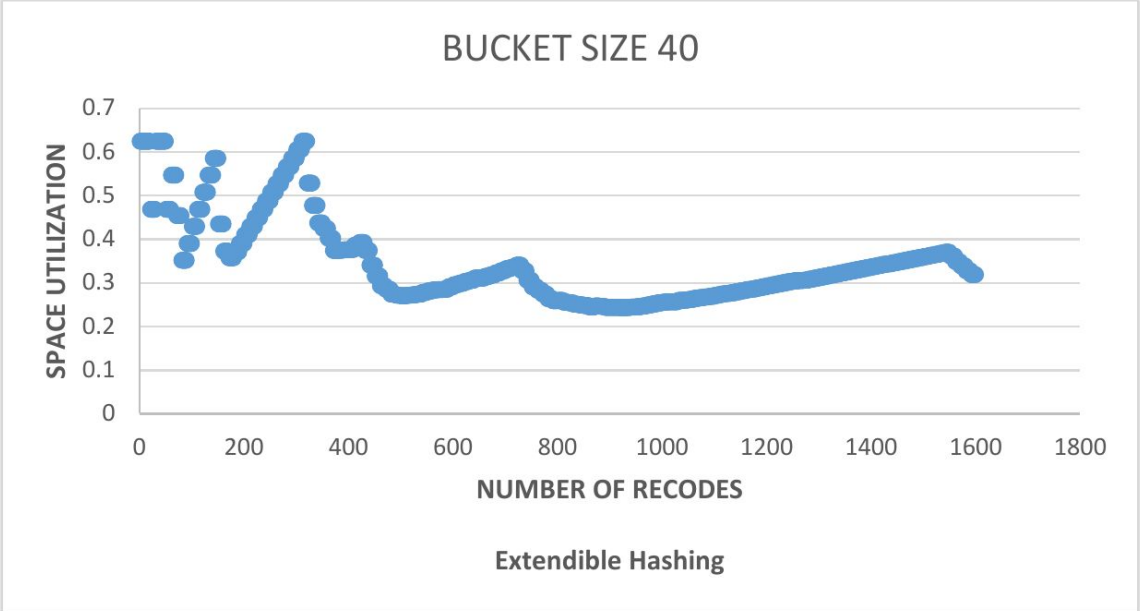
Comparing to Hashing method Extendible hashing have less cost w.r.t. Linear hashing. Since dataset is randomly generated rand follow Normal distribution, extendible hashing performs well.

Space Utilization:




BUCKET SIZE 40
SPACE UTILIZATION
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
1200
1400
1600
1800
NO OF RECORDS
Linear Hashing

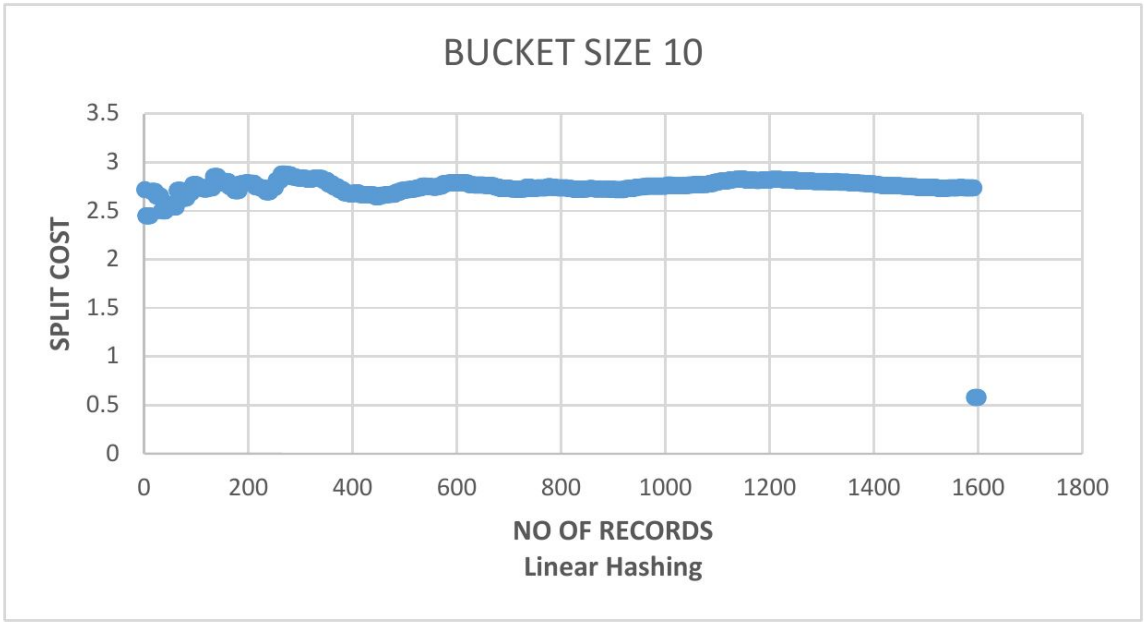


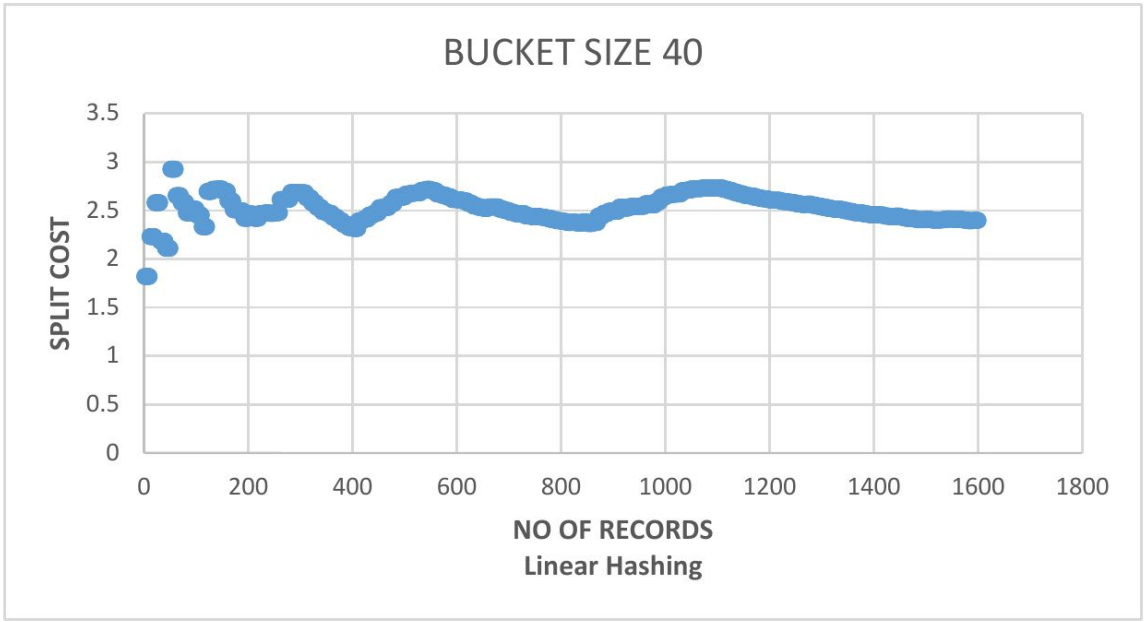


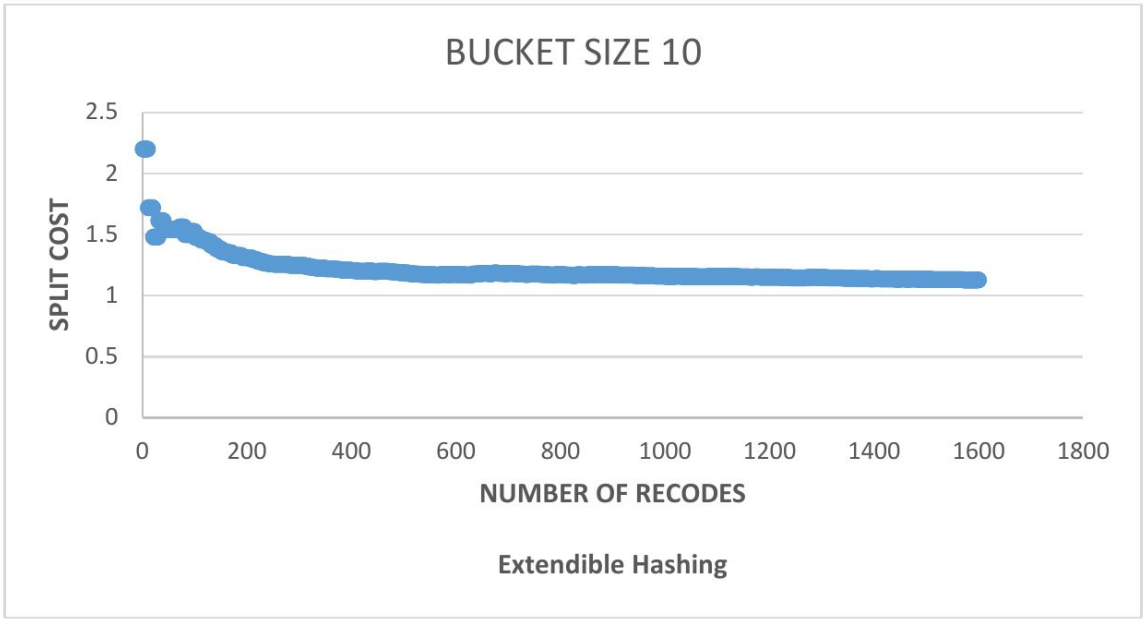
In space utilization clearly Linear hashing perform better than Extendible hashing. Since Linear hashing have no directory unlike extendible hashing, it has better space utilization.

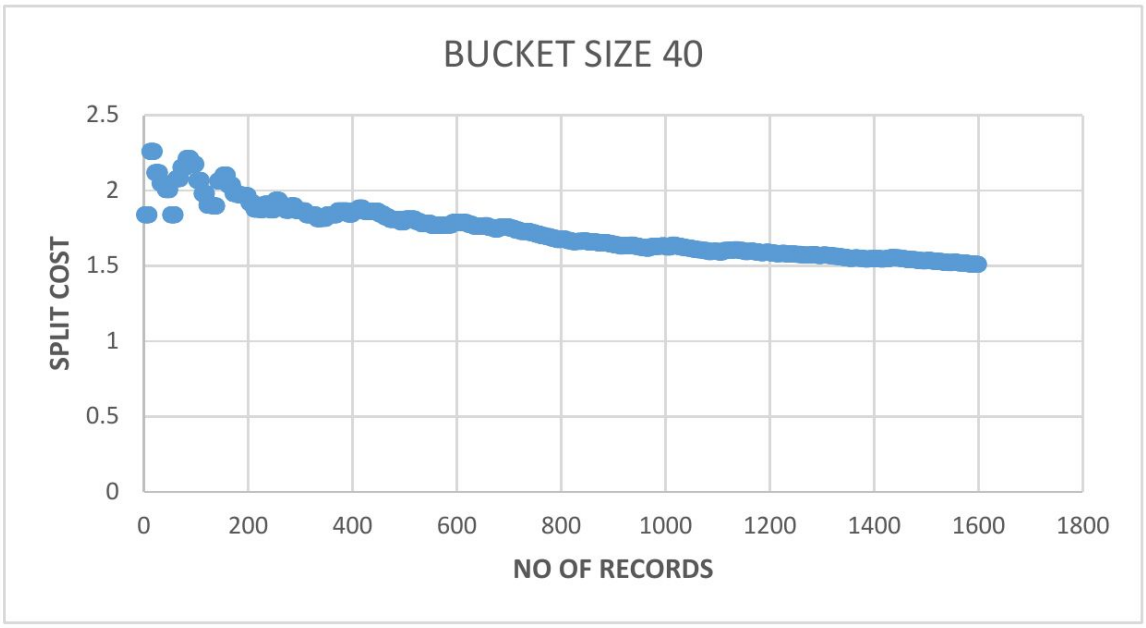
So, in application where memory resource are expensive linear hashing is prefered over extendible hashing.

Split Cost:









Looking at above plot splitting cost of linear Hashing is relatively higher than Extendible hashing. Reason for this could be, since extra read access is required to read buckets for split.

In Linear hashing split cost is almost same for both bucket size 10 and 40, while Extendible hashing have less split cost for bucket size 10 than of size 40.

Conclusion:

In extendible hashing, index grows smoothly with fewer page replace, but because of directory structure extra space and addition bucket access for insertion.
While Linear hashing avoids directory concepts but cause drastically switch in index function. (double or half).

Reference:

- http://www.iitrpr.ac.in/gunturi/courses/fall18/csl620/schedule.html
- https://www.csd.uoc.gr/~hy460/pdf/Performance%20comparison%20of%20extendible%20hashing%20and%20linear%20hashing%20techniques.pdf