Assignment 2 – Locality Sensitive Hashing

Output File –

```
////////. Locality Sensitive Hashing  ./////////


/******Get the input data and calculate its hash code and store in buckets****/

Input size 3

Input Data
[0.840188, 0.394383, 0.783099, 0.79844, 0.911647, 0.197551, 0.335223, 0.76823, 0.277775, 0.55397, 0.477397, 0.628871, 0.364784, 0.513401, 0.95223, 0.916
195]
[0.635712, 0.717297, 0.141603, 0.606969, 0.0163006, 0.242887, 0.137232, 0.804177, 0.156679, 0.400944, 0.12979, 0.108809, 0.998925, 0.218257, 0.512932, 0
.839112]
[0.61264, 0.296032, 0.637552, 0.524287, 0.493583, 0.972775, 0.292517, 0.771358, 0.526745, 0.769914, 0.400229, 0.891529, 0.283315, 0.352458, 0.807725, 0.
919026]

Projection vector size 2

Projection Data
[2.19173, 0.971841, -0.941838, -0.280727, 2.32506, 1.15326, 1.66495, 0.26878, -1.57542, 0.0901031, -0.571431, 1.09008, 1.16947, -0.068731, -0.114161, -0
.5746]
[0.435788, 0.934373, -0.295985, 1.34592, -0.561214, -0.253266, 0.578046, 1.13863, 0.536954, 0.738399, 0.257255, 0.28007, 1.43472, -0.858092, -0.189738,
-0.245128]


Dot Product Data
[4.15631, 2.61904]
[2.99552, 4.03353]
[3.5161, 2.51403]

Set of Similar value Data
[4, 3]
[3, 4]
[4, 3]


Set of Similar value Data
[43]
[34]
[43]


Normalize form of input Data
[1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]
[1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1]
[1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1]


Set of Similar value Data
[1.0111e+15]
[1.101e+15]
[1.01101e+15]


/******Partition the dataset into multiple buckets****/

bucket[34]--> 1.101e+15
bucket[43]--> 1.0111e+15  1.01101e+15

/******Check the generated query point and it's nearest cluster/bucket****/

Query Point Data
[0.447034, 0.226107, 0.187533, 0.276235, 0.556444, 0.416501, 0.169607, 0.906804, 0.103171, 0.126075, 0.495444, 0.760475, 0.984752, 0.935004, 0.684445, 0
.383188]


Dot Product for query_pt
[4.42931, 2.31081]


Set of Similar value Data for query_pt
[4, 2]


Set of Similar value Data
[42]


Normalize form of input Data
[0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0]


Set of Similar value query_pt Data
[1.0111e+15]


/******Search result for the calculated query points****/

Closest bucket[43] (Hash Code) is available for our Query Point Data-->42 (Hash Code) for query point --> [0.447034, 0.226107, 0.187533, 0.276235, 0.556
444, 0.416501, 0.169607, 0.906804, 0.103171, 0.126075, 0.495444, 0.760475, 0.984752, 0.935004, 0.684445, 0.383188]
```

Notes –

1. We have explained our code using comments in our c file. We have mentioned our comments for almost each method or statements in our code.
2. We haven't print any other values in the output such as mean value, variance value or tree values. Because it's taking so much space in the document as we have a lot of data. We are just printing Query points and Nearest neighbour point with their minimum distance value.

Steps -

1. We have generated input points of size 10000 with dimension 16.
2. We have used m=500 size vector which is calculated using gaussian factor chosen for LSH based on W scaling factor.
3. Then, we have calculated hash code for each input by taking dot product of the input data and m factor hashing values.
4. We have merged all hash code keys along with their value in respective buckets/hash tables.
5. We took one random query point to search the nearest or exact cluster. We calculated hash code for query point and then check in the bucket and return the nearest bucket.

Thank you for such thoughtful assignment.