COL 819 : ADVANCED DISTRIBUTED SYSTEM (Report)

csz208507 (Rajat Singh)

March 2021

1 Introduction

In this assignment I have implemented/simulated a Pastry network. The implementation has some key aspects of the algorithm such as the addition of nodes, deletion of nodes, and look up queries for data.

2 How to run the code

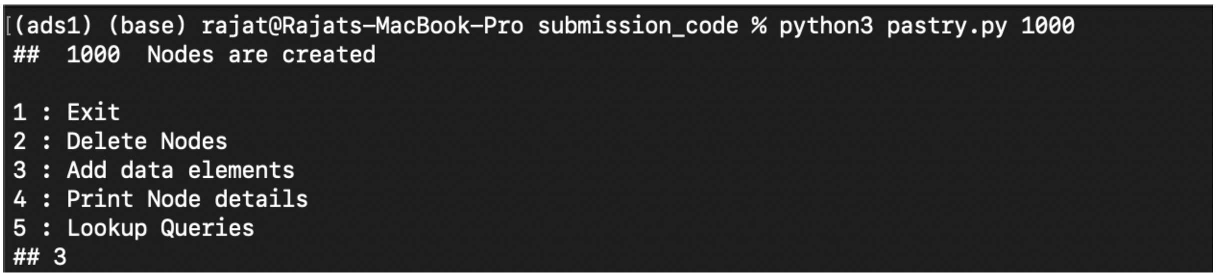
To run the code use the command given below :

```
python3 pastry.py <Number_of_nodes>
```

In the above command the $< Number\_of\_nodes >$ is the count of nodes we want to create in the Pastry network. If we want to create 1000 nodes then we have to use the command given below.

```
python3 pastry.py 1000
```

Once we run the above command 1000 nodes will be created in the Pastry network and an interactive menu will be displayed as shown below.



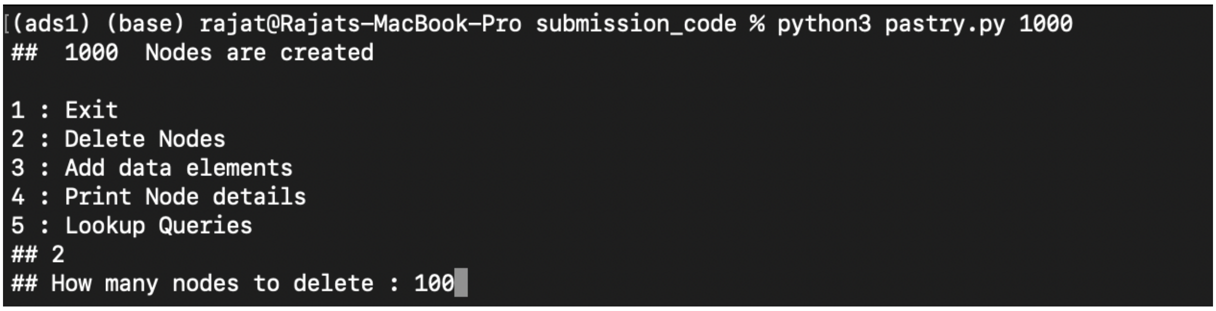
```
[(ads1) (base) rajat@Rajats-MacBook-Pro submission_code % python3 pastry.py 1000
##  1000  Nodes are created

1 : Exit
2 : Delete Nodes
3 : Add data elements
4 : Print Node details
5 : Lookup Queries
## 3
```

Figure 1: Menu

In Figure 1 we have 5 options as follows

1. **To exit from the simulation** : If we press key 1 then the simulation will be stopped.

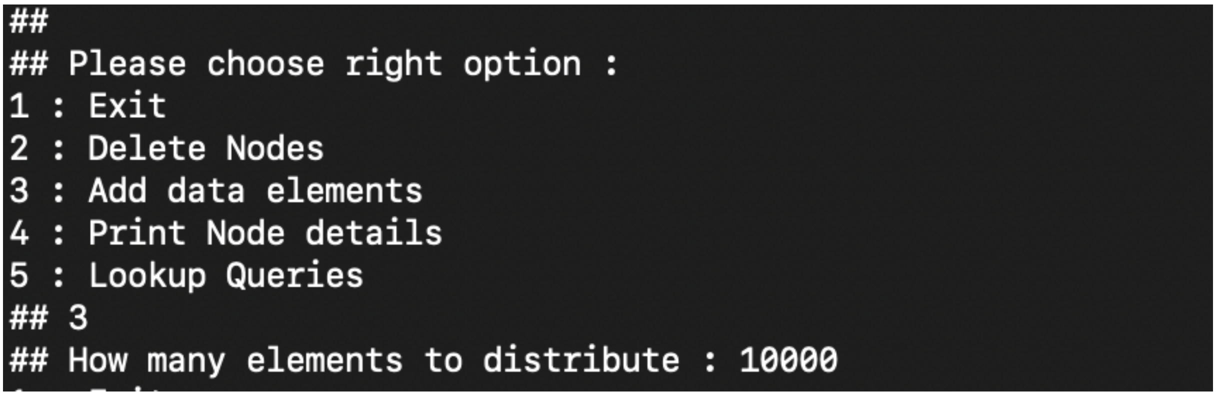
2. **To Delete nodes** : If we press key 2 then the simulation will ask for the number of the nodes we want to delete. If the number of nodes to be deleted are more than the number of existing nodes then it will throw an error message (Figure 2).



```
[(ads1) (base) rajat@Rajats-MacBook-Pro submission_code % python3 pastry.py 1000
##  1000  Nodes are created

1 : Exit
2 : Delete Nodes
3 : Add data elements
4 : Print Node details
5 : Lookup Queries
## 2
## How many nodes to delete : 100
```

Figure 2: Delete Nodes

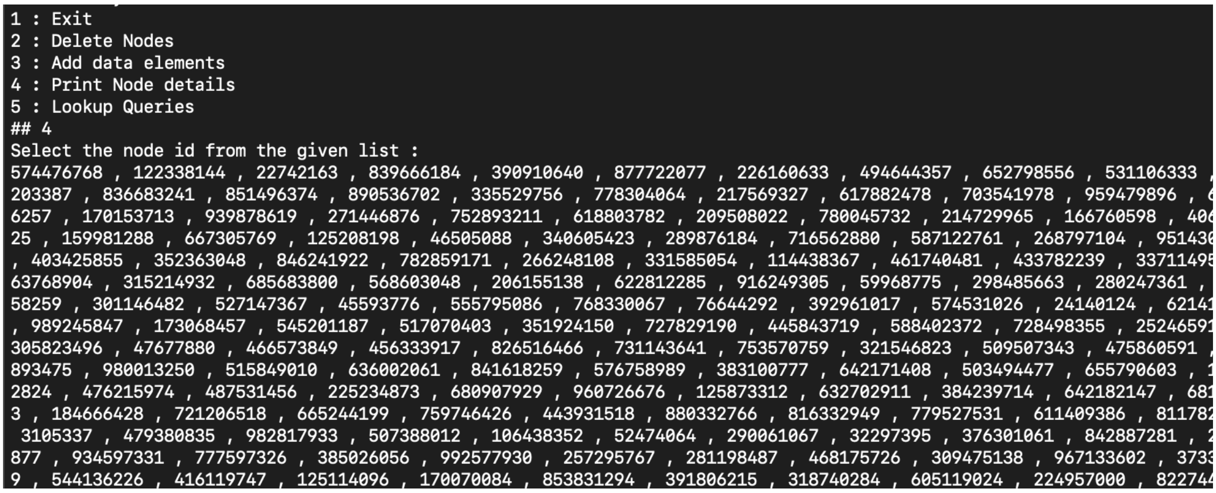
3. **To add data elements** : If we press key 3 then we can add the data elements to the nodes randomly as shown below in the Figure 3.



```
##
## Please choose right option :
1 : Exit
2 : Delete Nodes
3 : Add data elements
4 : Print Node details
5 : Lookup Queries
## 3
## How many elements to distribute : 10000
```

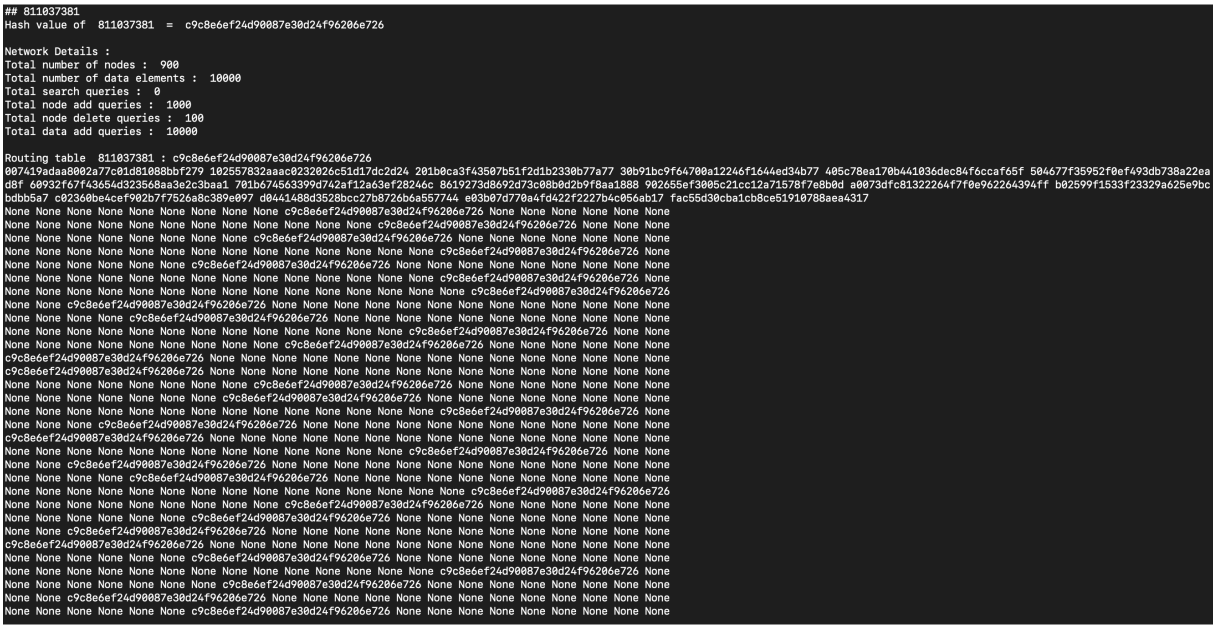
Figure 3: Add data elements

4. **To Print Node details** : If we press key 4 then we can see the list of all nodes in the network and we can select any node to see detail of a particular node as shown in the figure 4.



```
1 : Exit
2 : Delete Nodes
3 : Add data elements
4 : Print Node details
5 : Lookup Queries
## 4
Select the node id from the given list :
574476768 , 122338144 , 22742163 , 839666184 , 390910640 , 877722077 , 226160633 , 494644357 , 652798556 , 531106333 ,
203387 , 836683241 , 851496374 , 890536702 , 335529756 , 778304064 , 217569327 , 617882478 , 703541978 , 959479896 , 6
6257 , 170153713 , 939878619 , 271446876 , 752893211 , 618803782 , 209508022 , 780045732 , 214729965 , 166760598 , 406
25 , 159981288 , 667305769 , 125208198 , 46505088 , 340605423 , 289876184 , 716562880 , 587122761 , 268797104 , 951430
, 403425855 , 352363048 , 846241922 , 782859171 , 266248108 , 331585054 , 114438367 , 461740481 , 433782239 , 33711495
63768904 , 315214932 , 685683800 , 568603048 , 206155138 , 622812285 , 916249305 , 59968775 , 298485663 , 280247361 ,
58259 , 301146482 , 527147367 , 45593776 , 555795086 , 768330067 , 76644292 , 392961017 , 574531026 , 24140124 , 62141
, 989245847 , 173068457 , 545201187 , 517070403 , 351924150 , 727829190 , 445843719 , 588402372 , 728498355 , 25246591
305823496 , 47677880 , 466573849 , 456333917 , 826516466 , 731143641 , 753570759 , 321546823 , 509507343 , 475860591 ,
893475 , 980013250 , 515849010 , 636002061 , 841618259 , 576758989 , 383100777 , 642171408 , 503494477 , 655790603 , 1
2824 , 476215974 , 487531456 , 225234873 , 680907929 , 960726676 , 125873312 , 632702911 , 384239714 , 642182147 , 681
3 , 184666428 , 721206518 , 665244199 , 759746426 , 443931518 , 880332766 , 816332949 , 779527531 , 611409386 , 811782
 3105337 , 479380835 , 982817933 , 507388012 , 106438352 , 52474064 , 290061067 , 32297395 , 376301061 , 842887281 , 2
877 , 934597331 , 777597326 , 385026056 , 992577930 , 257295767 , 281198487 , 468175726 , 309475138 , 967133602 , 3733
9 , 544136226 , 416119747 , 125114096 , 170070084 , 853831294 , 391806215 , 318740284 , 605119024 , 224957000 , 822744
```

Figure 4: Print node details

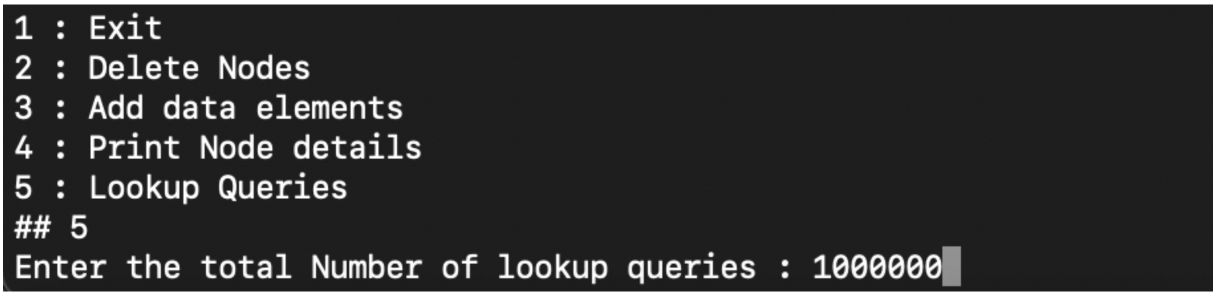


```
## 811037381
Hash value of  811037381  =  c9c8e6ef24d90087e30d24f96206e726

Network Details :
Total number of nodes :  900
Total number of data elements :  10000
Total search queries :  0
Total node add queries :  1000
Total node delete queries :  100
Total data add queries :  10000

Routing table  811037381 : c9c8e6ef24d90087e30d24f96206e726
007419adaa8002a77c01d81088bbf279 102557832aaac0232026c51d17dc2d24 201b0ca3f43507b51f2d1b2330b77a77 30b91bc9f64700a12246f1644ed34b77 405c78ea170b441036dec84f6ccaf65f 504677f35952f0ef493db738a22ea
d8f 60932f67f43654d323568aa3e2c3baa1 701b674563399d742af12a63ef28246c 8619273d8692d73c08b0d2b9f8aa1888 902655ef3005c21cc12a71578f7e8b0d a0073dfc81322264f7f0e962264394ff b02599f1533f23329a625e9bc
bdbb5a7 c02360be4cef902b7f7526a8c389e097 d0441488d3528bcc27b8726b6a557744 e03b07d770a4fd422f2227b4c056ab17 fac55d30cba1cb8ce51910788aea4317
None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None
None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None
None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None
None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None
None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None
None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None
None None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726
None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None
None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None
None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None
None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None
c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None None None
c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None None None
None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None
None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None
None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None
None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None
c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None None None
None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None
None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None
None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None
None None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726
None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None
None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None
None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None
c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None None None
None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None
None None None None None None None None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None
None None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None
None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None None None None None
None None None None None None c9c8e6ef24d90087e30d24f96206e726 None None None None None None None None None
```

Figure 5: Node Details

Here in the above table I have used the hash values as the node id for all the nodes. The above table is 32 * 16.

5. **Lookup Queries** : If we press key 4 then we can run the lookup command by giving the number of random look-ups we need to do as shown in the figure given below.



```
1 : Exit
2 : Delete Nodes
3 : Add data elements
4 : Print Node details
5 : Lookup Queries
## 5
Enter the total Number of lookup queries : 1000000
```

Figure 6: Look-up Queries

3 Evaluation

Case 1 :

Configuration for this case is :
1. Number of nodes = 100
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :

Average number of hops for a search query = 1.949261 hops

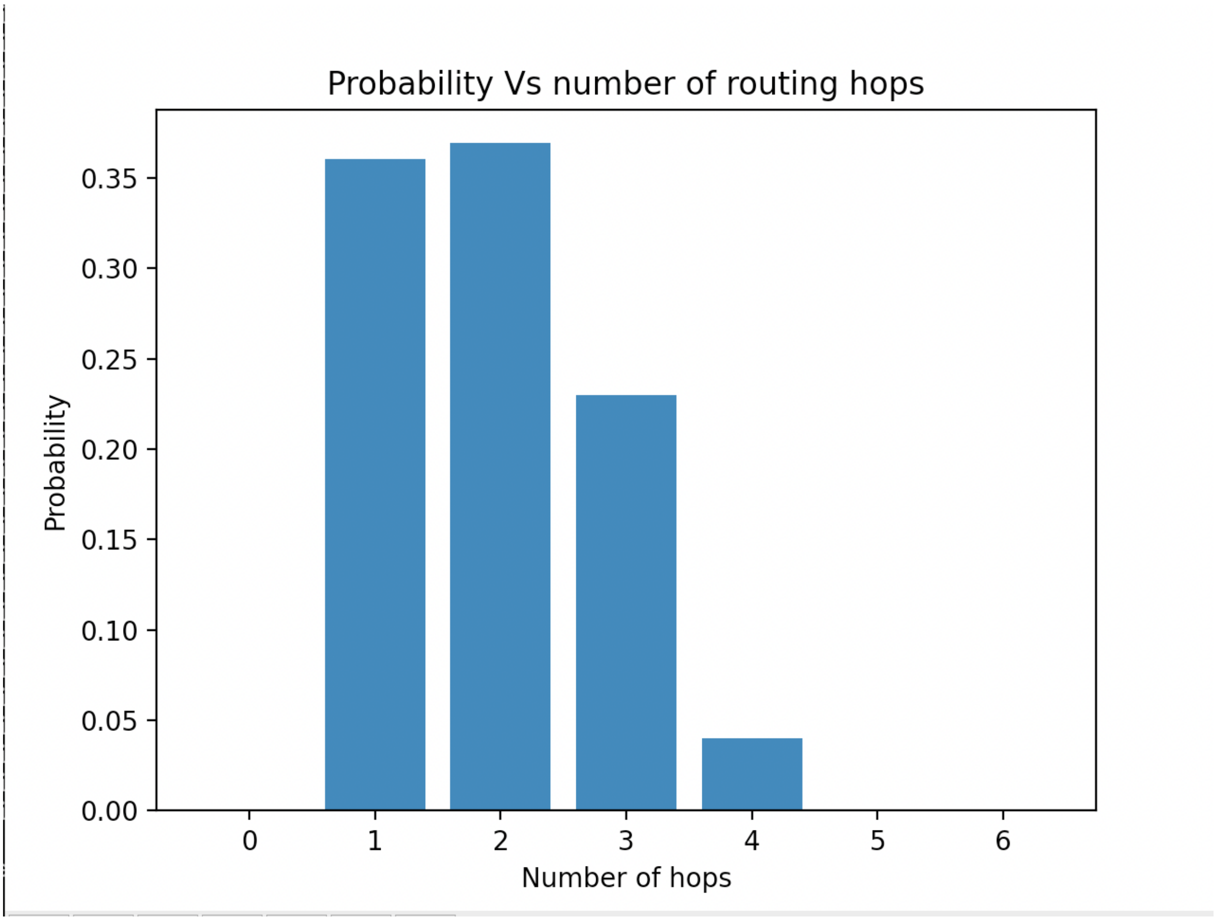


Figure 7: Probability versus number of routing hops, b=4, $|L| = 16$, $|M| = 32$, nodes = 100 , lookups = 1000000

We are getting this trend where Average number of hops for a search query is approx 1.9 hops. About 24 percent of the queries take 3 hops to reach to the destination. About 36 percent of the queries take 2 hops to reach to the destination node.About 35 percent of the queries take 1 hops to reach to the destination node.

The above trend is valid as we have less number of nodes thus we can reach to the other node in few number of steps as number of hops are directly proportional to log of the number of nodes.

Here we are getting decent amount of value at number of hops = 3. The main reason for this could be the random function using which I am distributing the data-points as it could be distributing the values such that we are getting more of 3 hops for searching a query.

Now we will randomly delete 50 percent of the nodes and distribute the data-points to the remaining nodes.

Configuration for this case is :

1. Number of nodes = 50
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :

Average number of hops for a search query = 1.48442 hops

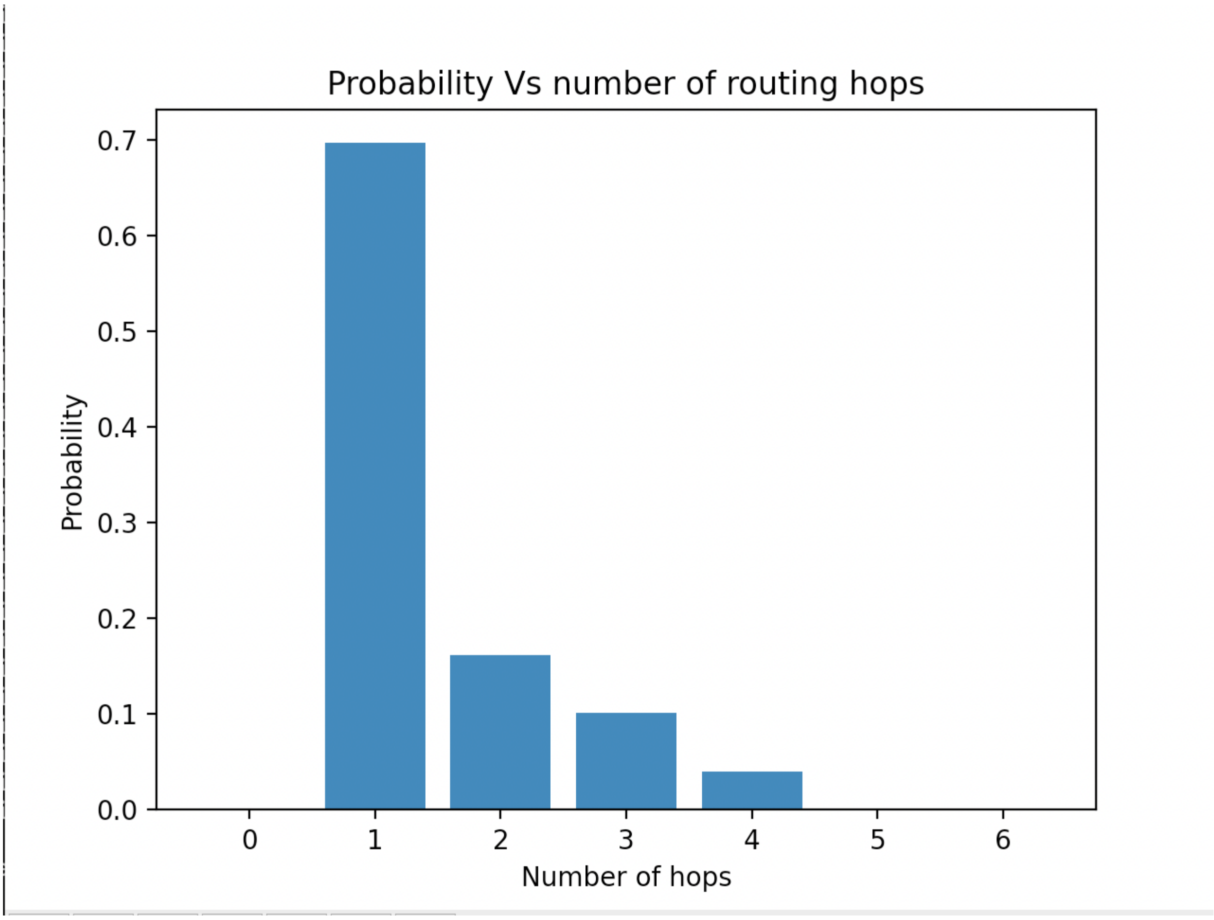


Figure 8: Probability versus number of routing hops, b=4, $|L| = 16$, $|M| = 32$, nodes = 50 , lookups = 1000000

I am getting this trend where Average number of hops for a search query is approx 1.48 hops. About 70 percent of the queries take 1 hops to reach to the destination. About 18 percent of the queries take 2 hops to reach to the destination node.

The above trend is valid as we have less number of nodes thus we can reach to the other node in few number of steps as number of hops are directly proportional to log of the number of nodes.

We can also see that as the number of nodes decreases the average number of nodes for a search query increases.

Case 2 :
Configuration for this case is :
1. Number of nodes = 500
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :
Average number of hops for a search query = 2.516946 hops

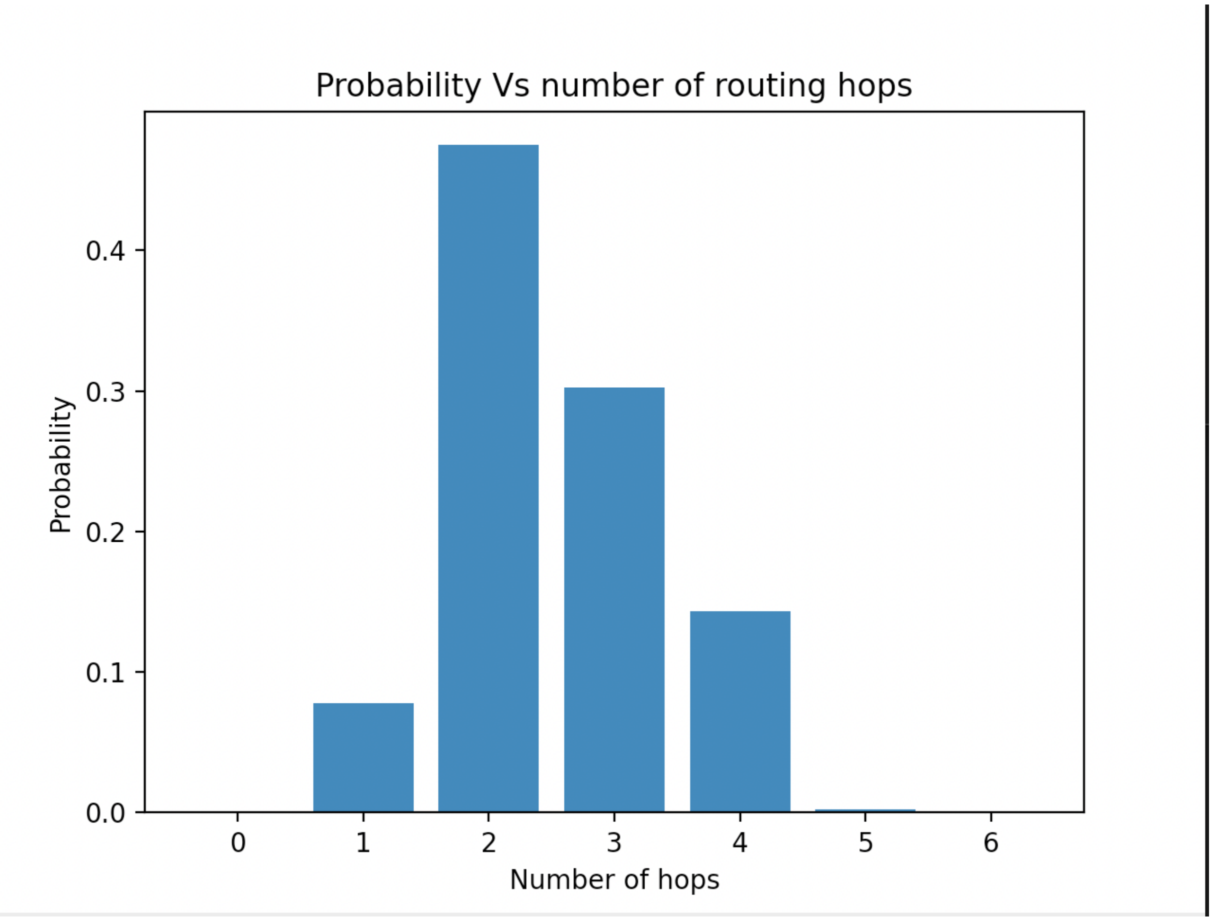


Figure 9: Probability versus number of routing hops, b=4, $|L|$ = 16, $|M|$ = 32, nodes = 500 , lookups = 1000000

I am getting this trend where Average number of hops for a search query is approx 2.51 hops. About 48 percent of the queries take 2 hops to reach to the destination. About 30 percent of the queries take 3 hops to reach to the destination node.

We can also see that as the number of nodes increases the average number of nodes for a search query increases as the distribution of the data-points is sparse.

Now we will randomly delete 50 percent of the nodes and distribute the data-points to the remaining nodes.

Configuration for this case is :

1. Number of nodes = 250
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :

Average number of hops for a search query = 2.278711 hops

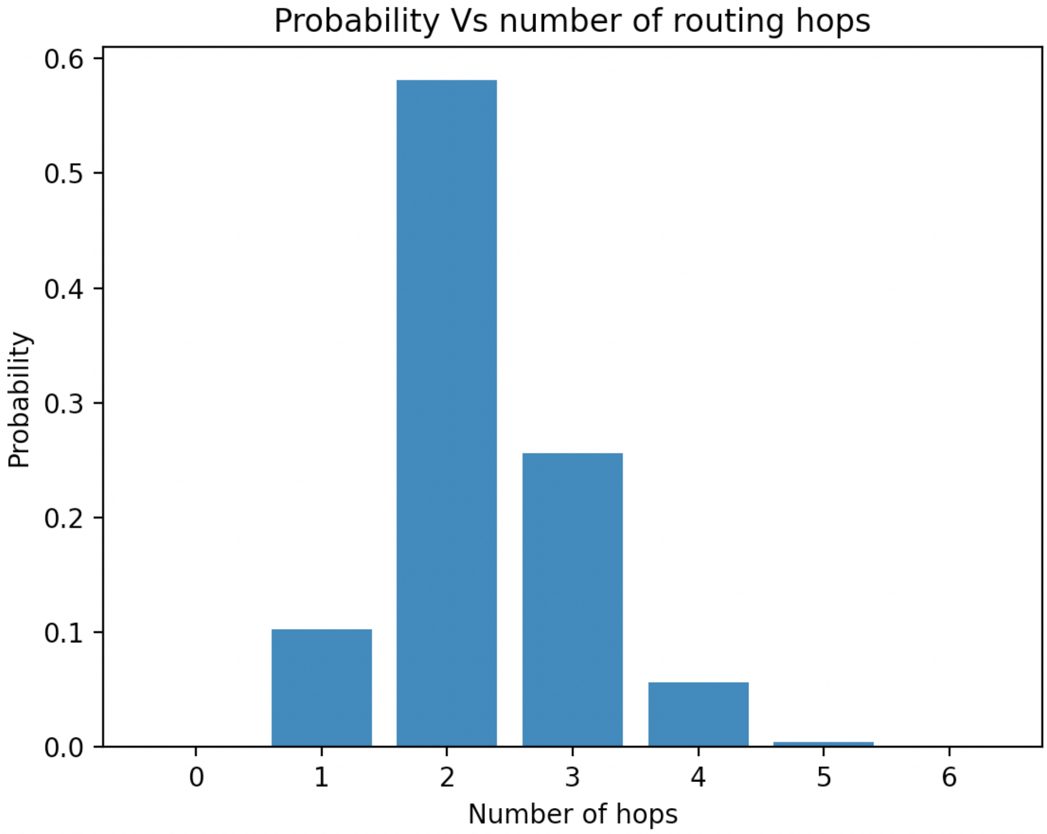


Figure 10: Probability versus number of routing hops, b=4, $|L| = 16$, $|M| = 32$, nodes = 250 , lookups = 1000000

I am getting this trend where Average number of hops for a search query is approx 2.27 hops. About 58 percent of the queries take 2 hops to reach to the destination. About 25 percent of the queries take 3 hops to reach to the destination node.

Now again when we decreased the number of nodes the average number of hops required for a search query is also decreased.

Case 3:
Configuration for this case is :
1. Number of nodes = 1000
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :
Average number of hops for a search query = 2.44616 hops

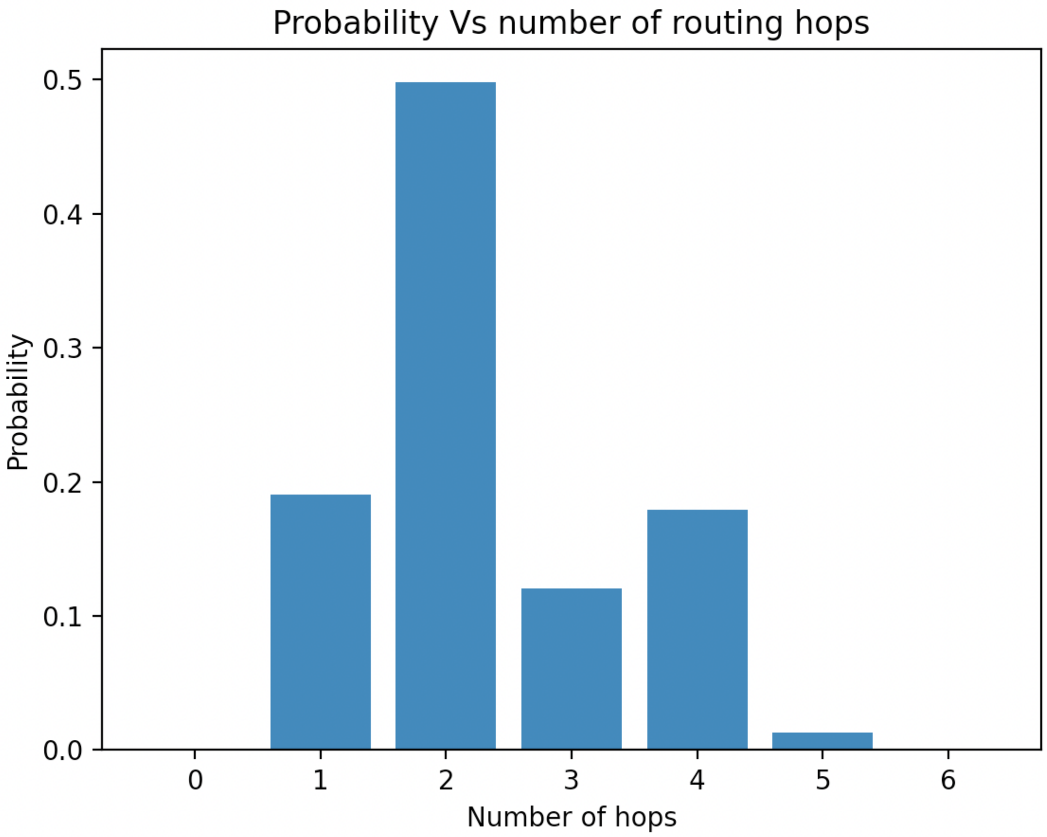


Figure 11: Probability versus number of routing hops, b=4, $|L| = 16$, $|M| = 32$, nodes = 1000 , lookups = 1000000

I am getting this trend where Average number of hops for a search query is approx 2.44 hops. About 50 percent of the queries take 2 hops to reach to the destination. About 19 percent of the queries take 3 hops to reach to the destination node. About 17 percent of the queries take 1 hops to reach to the destination node.

It is little less than what is mentioned in the Pastry paper and I think i am getting this result because of the random generator function which is distributing the data points to the respective nodes.

Now we will randomly delete 50 percent of the nodes and distribute the data-points to the remaining nodes.

Configuration for this case is :

1. Number of nodes = 500
2. Data-points = 10000
3. Search Queries/ Look-ups = 1 million

Result :

Average number of hops for a search query = 1.909299 hops

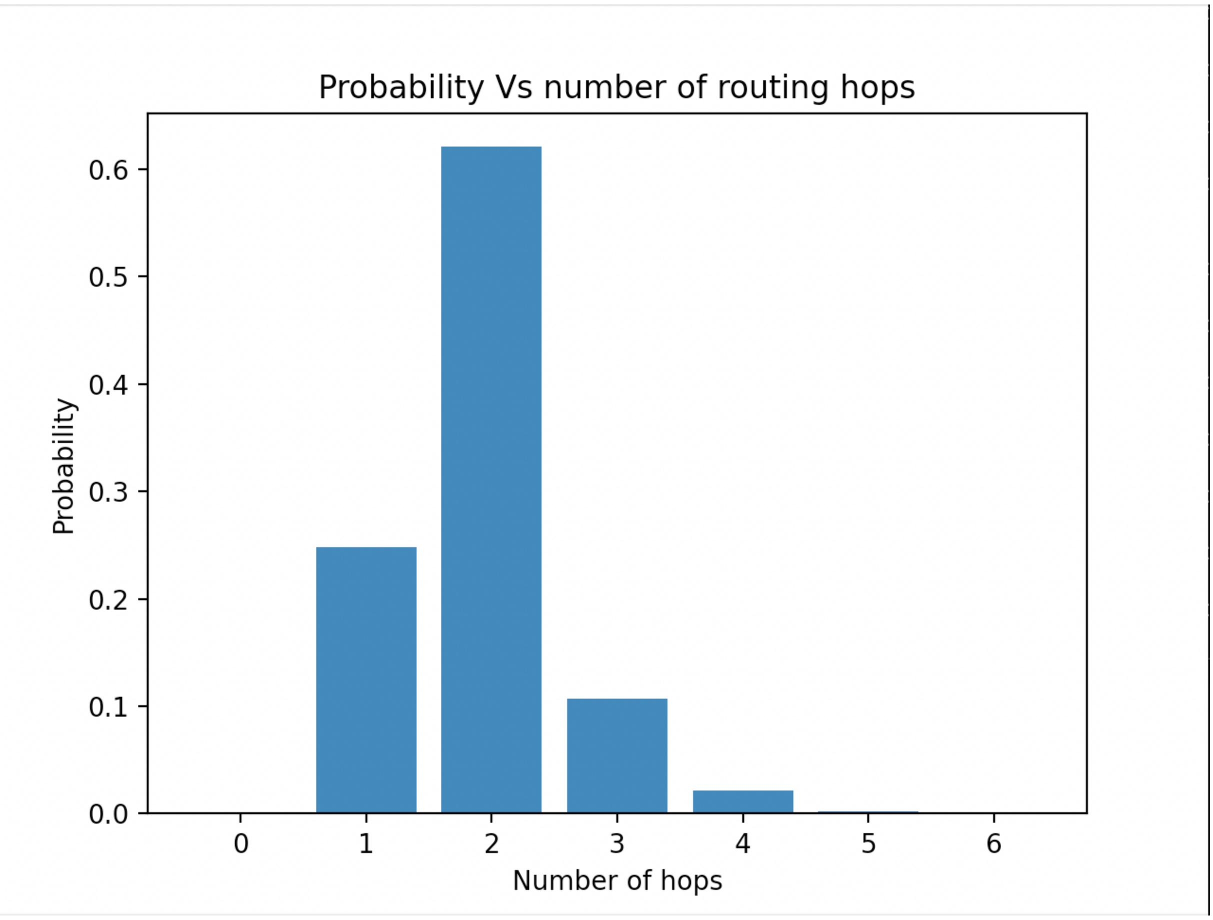
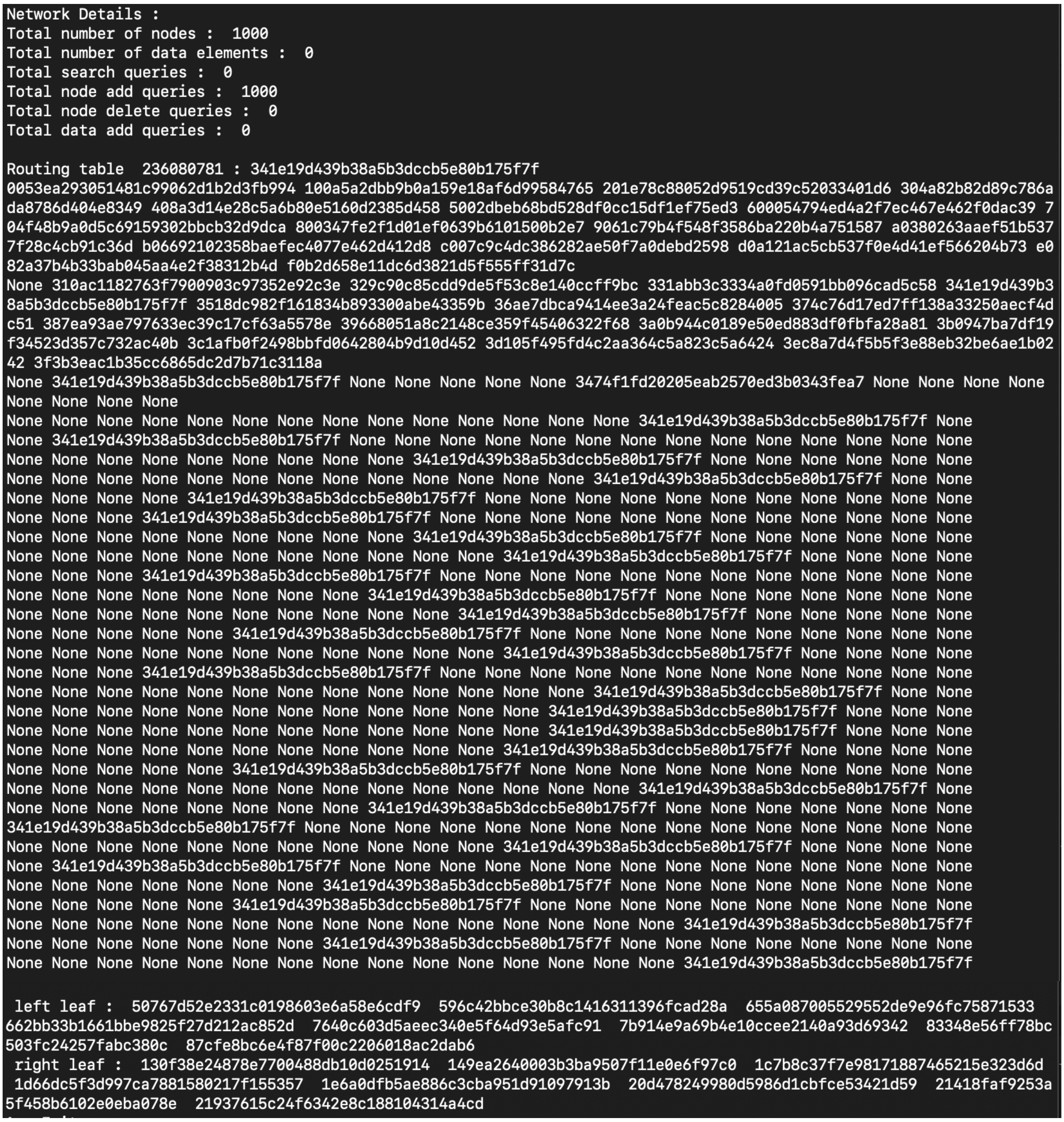


Figure 12: Probability versus number of routing hops, b=4, $|L| = 16$, $|M| = 32$, nodes = 500 , lookups = 1000000

I am getting this trend where Average number of hops for a search query is approx 1.91 hops. About 62 percent of the queries take 2 hops to reach to the destination. About 26 percent of the queries take 1 hops to reach to the destination node. About 11 percent of the queries take 3 hops to reach to the destination node.

As we have decreased the number of nodes thus the average number of hops needed to search a query is also decreased.

4 Node Details



```
Network Details :
Total number of nodes :  1000
Total number of data elements :  0
Total search queries :  0
Total node add queries :  1000
Total node delete queries :  0
Total data add queries :  0

Routing table  236080781 : 341e19d439b38a5b3dccb5e80b175f7f
0053ea293051481c99062d1b2d3fb994 100a5a2dbb9b0a159e18af6d99584765 201e78c88052d9519cd39c52033401d6 304a82b82d89c786a
da8786d404e8349 408a3d14e28c5a6b80e5160d2385d458 5002dbeb68bd528df0cc15df1ef75ed3 600054794ed4a2f7ec467e462f0dac39 7
04f48b9a0d5c69159302bbcb32d9dca 800347fe2f1d01ef0639b6101500b2e7 9061c79b4f548f3586ba220b4a751587 a0380263aaef51b537
7f28c4cb91c36d b06692102358baefec4077e462d412d8 c007c9c4dc386282ae50f7a0debd2598 d0a121ac5cb537f0e4d41ef566204b73 e0
82a37b4b33bab045aa4e2f38312b4d f0b2d658e11dc6d3821d5f555ff31d7c
None 310ac1182763f7900903c97352e92c3e 329c90c85cdd9de5f53c8e140ccff9bc 331abb3c3334a0fd0591bb096cad5c58 341e19d439b3
8a5b3dccb5e80b175f7f 3518dc982f161834b893300abe43359b 36ae7dbca9414ee3a24feac5c8284005 374c76d17ed7ff138a33250aecf4d
c51 387ea93ae797633ec39c17cf63a5578e 39668051a8c2148ce359f45406322f68 3a0b944c0189e50ed883df0fbfa28a81 3b0947ba7df19
f34523d357c732ac40b 3c1afb0f2498bbfd0642804b9d10d452 3d105f495fd4c2aa364c5a823c5a6424 3ec8a7d4f5b5f3e88eb32be6ae1b02
42 3f3b3eac1b35cc6865dc2d7b71c3118a
None 341e19d439b38a5b3dccb5e80b175f7f None None None None None 3474f1fd20205eab2570ed3b0343fea7 None None None None
None None None None
None None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None
None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None None None
None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None
None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None
None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None
None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None
None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None
None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None
None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None
None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None
None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None
None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None
None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None
None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None
None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None
None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None
None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None
None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None
None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None
None None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None
None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None
341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None None None None
None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None
None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None None None None None
None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None
None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None None None
None None None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f
None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f None None None None None None None None
None None None None None None None None None None None None None None None 341e19d439b38a5b3dccb5e80b175f7f


 left leaf :  50767d52e2331c0198603e6a58e6cdf9  596c42bbce30b8c1416311396fcad28a  655a087005529552de9e96fc75871533
662bb33b1661bbe9825f27d212ac852d  7640c603d5aeec340e5f64d93e5afc91  7b914e9a69b4e10ccee2140a93d69342  83348e56ff78bc
503fc24257fabc380c  87cfe8bc6e4f87f00c2206018ac2dab6
 right leaf :  130f38e24878e7700488db10d0251914  149ea2640003b3ba9507f11e0e6f97c0  1c7b8c37f7e98171887465215e323d6d
 1d66dc5f3d997ca7881580217f155357  1e6a0dfb5ae886c3cba951d91097913b  20d478249980d5986d1cbfce53421d59  21418faf9253a
5f458b6102e0eba078e  21937615c24f6342e8c188104314a4cd
```

Figure 13: Node Details

5 Conclusion

For creating unique node id, I am using a build in random function which generates a unique id and its position in 2D plain. Thus each time when I am running the simulation it will new set of random nodes thus the arrangement of nodes will be different in each run. I have created the data elements randomly and distributed it to the nodes thus the average number of hops required to search a data element depends upon the distribution thus at every run the average number of hops required will be different.
After all the experiments we can conclude that if we are decreasing the number of nodes then we need less number of average number of hops required to search for a data element in the network. The average number of search required is between 1.4 to 2.6 which is quiet obvious as we are using very less number of nodes i.e in range of 50 to 1000.