CSE416A hw1

Yushu Liu

September 24, 2018

CSE416A: hw 1
Yushu Liu, 457978

---

1 Question 1

1.1 (a)

For a complete graph, the number of edges is:

$$m = \frac{n(n-1)}{2}$$

1.2 (b)

Average node degree is n-1 for a complete graph with n nodes.

1.3 (c)

We could use weighted graph to model the interaction of people as following:
Setting up a complete graph where each nodes represents one person and one edge represents the degree of interaction between people. Using data from facebook as example, to model the data we could set the weight of edge between two people(nodes) to be the inverse of number of common friends between them. If two people have no common friends, the weight will be infinite (inverse of 0). The result graph will be fully connected graph with people closer to each other has "shorter" distance between them.

1.4 (d)

Yes.

Proof. We could prove the statement by contradiction. Say if there is some path form u to w, S, is shorter than Q, and let L denote the segment of P between w and v. Then we can find another path from u and v shorter than P since $S < Q$, so $S + L < Q + L = P$, which contradicts with the fact that P is the shortest path.
Therefore Q has to be the shortest path between u and w. □

---

2 Question 2

2.1 (a)

The following table is the summary of BFS performed on the graph:

Node	Current Source	Case	CARN	LINC	HARV	BBN	MIT	UTAH	RAND	SDC	SRI	UCLA	STAN	UCSB
Status after visit 0	-	0	N	N	N	N	N	N	N	N	N	N	N	N
Status after visit 1	CASE	0	1	1	N	N	N	N	N	N	N	N	N	N
Status after visit 2	LINC, CARN	0	1	1	2	N	2	N	N	N	N	N	N	N
Status after visit 3	HARV, MIT	0	1	1	2	3	2	3	N	N	N	N	N	N
Status after visit 4	BBN, UTAH	0	1	1	2	3	2	3	4	4	4	N	N	N
Status after vist 5	RAND, SDC, UCLA	0	1	1	2	3	2	3	4	4	4	5	5	5

First, setting source to Case and set distance to all other colleges to NULL (represented by N). For each visit start from visit 1, set current source to the neighbors of previous source, and replace distance of all neighbors has distance "Null" of current source by distance of current source + 1. The algorithm is completed after 6 iteration (given that all Null are replaced by numerical value) and the number in last row represents the distance to Case for all vertices on graph.

BFS can only be applied to unweighted graph and DIJKSTRA can be applied to weighted graph.

2.2 (b)

The screen shot attached includes the edge list and the output of DIJKSTRA algorithm on Arpanet graph.

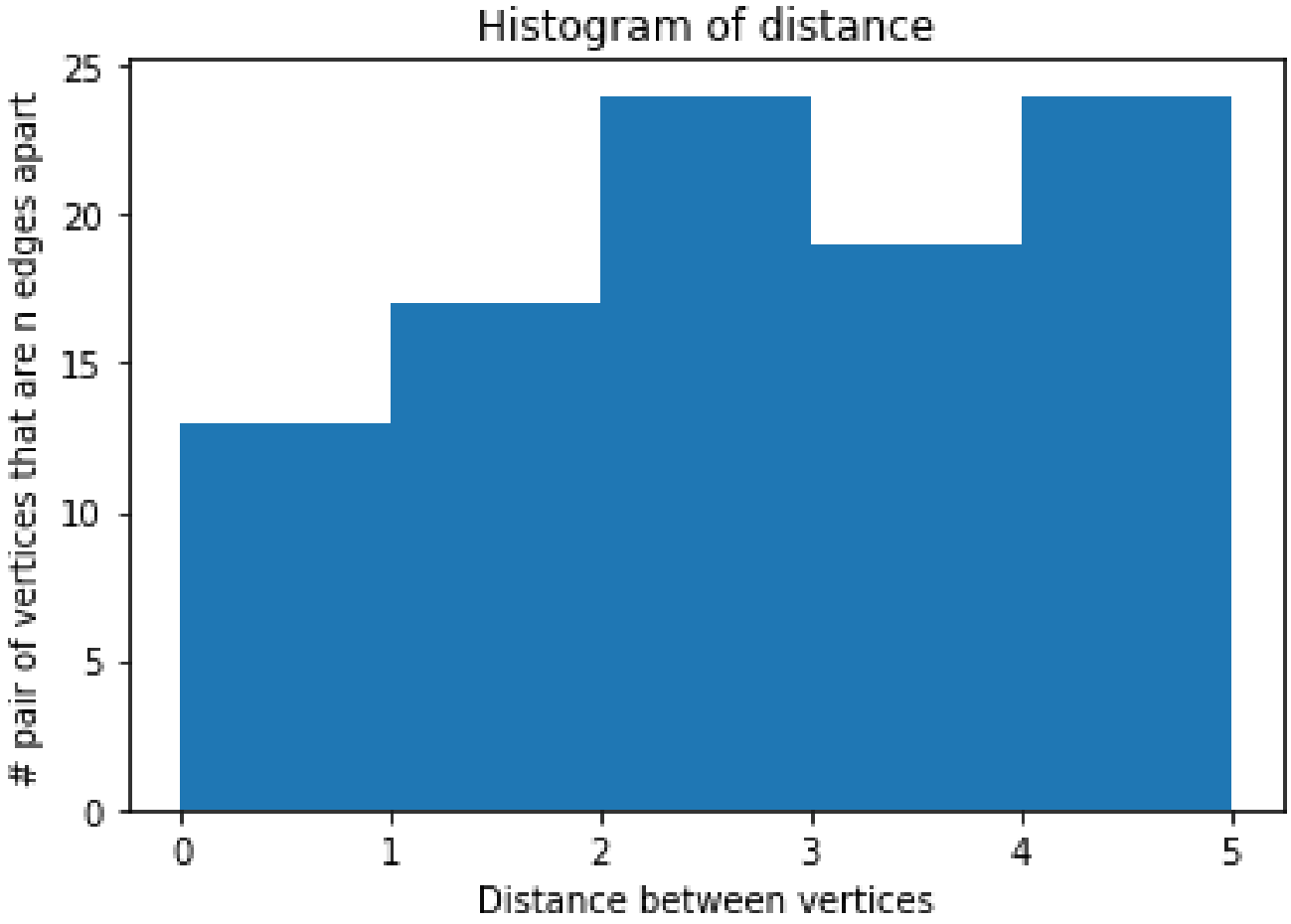
```
In [105]: G.edges()
```

```
Out[105]: EdgeView([('CASE', 'CRAN'), ('CASE', 'LINC'), ('CRAN', 'HARV'), ('LINC', 'MIT'), ('HARV', 'BBN'), ('BBN', 'MIT'), ('BBN', 'RAN
          D'), ('MIT', 'UTAH'), ('UTAH', 'SDC'), ('UTAH', 'SRI'), ('RAND', 'SDC'), ('RAND', 'UCLA'), ('SRI', 'STAN'), ('SRI', 'UCLA'),
          ('SRI', 'UCSB'), ('UCLA', 'STAN'), ('UCLA', 'UCSB')])
```

```
In [106]: result = nx.shortest_path_length(G, source = "CASE")
          import operator
          sorted_result = sorted(result.items(), key = operator.itemgetter(1))
          sorted_result
```

```
Out[106]: [('CASE', 0),
           ('CRAN', 1),
           ('LINC', 1),
           ('HARV', 2),
           ('MIT', 2),
           ('BBN', 3),
           ('UTAH', 3),
           ('RAND', 4),
           ('SDC', 4),
           ('SRI', 4),
           ('UCLA', 5),
           ('STAN', 5),
           ('UCSB', 5)]
```

2.3 (c)


Histogram of distance
pair of vertices that are n edges apart
Distance between vertices
0
5
10
15
20
25
0
1
2
3
4
5

2.3.1 3

See code submission.