Homework #10

Graphs III

Anastasiia Konoplina

---

*1. Create an adjacency matrix representation G of this graph. Calculate G*G. Draw the respective result.*

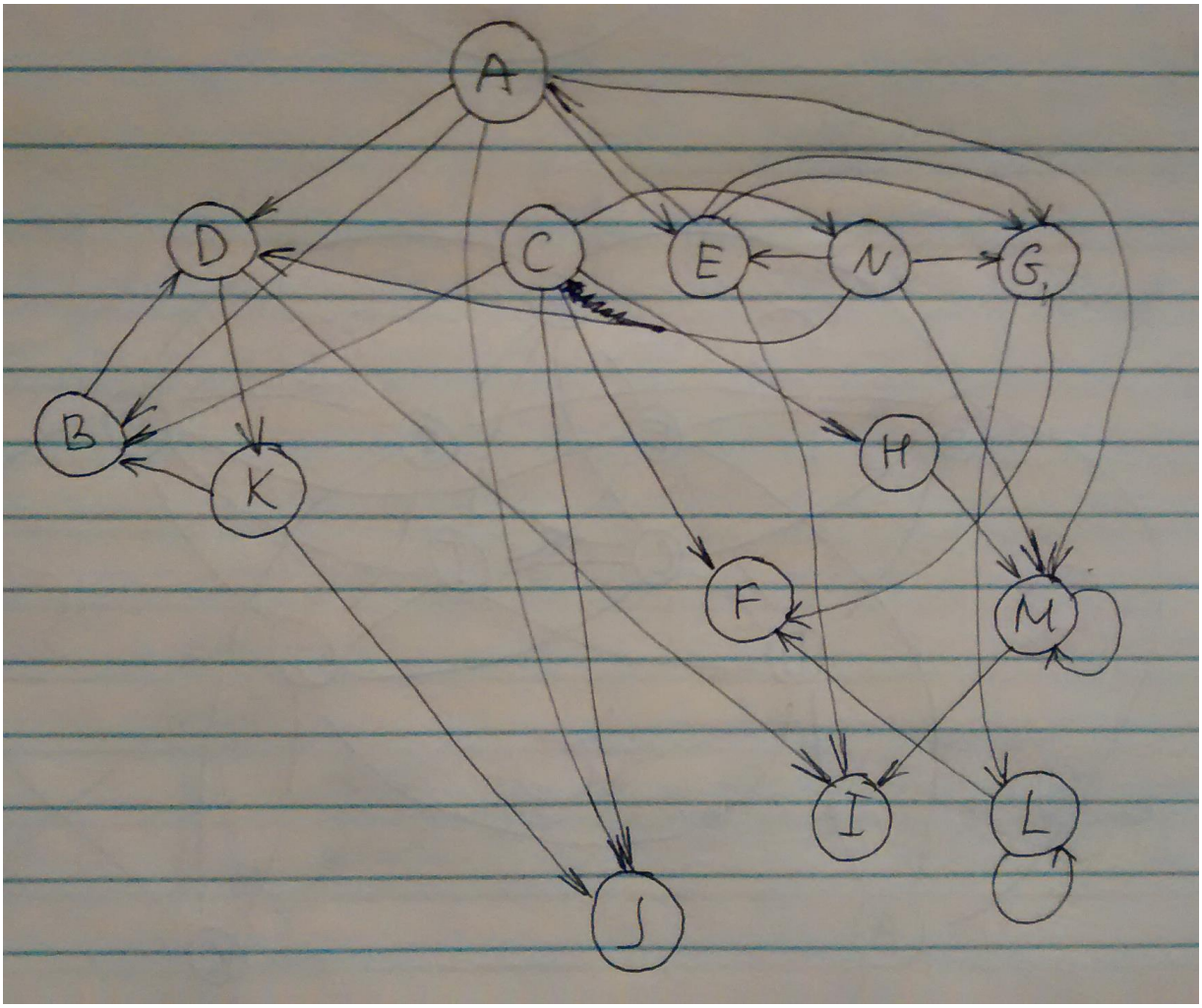
Adjacency matrix:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	0	1	1	0	0	1	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	1	0	0	0
C	0	0	0	1	1	0	0	0	0	0	0	0	0	0
D	0	1	0	0	0	0	0	0	0	1	0	0	0	0
E	0	0	0	0	0	1	0	1	0	0	0	0	0	1
F	0	0	0	0	0	0	0	0	1	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	1	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	1	0	0	0	0	0
K	0	0	0	1	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	1	0
M	0	0	0	0	0	1	0	0	0	0	0	1	0	0
N	1	0	0	0	0	0	1	0	0	0	0	0	0	0

G*G (I put zeros as diagonal elements):

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	1	0	1	1	0	0	0	0	1	0	0	1	0
B	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	1	0	1	0	1	0	0	0	1
D	0	0	0	0	0	0	0	0	1	0	1	0	0	0
E	1	0	0	0	0	0	2	0	1	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	1	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	1	0	0	0	0	0	0	0	1	0	0	0	0
L	0	0	0	0	0	1	0	0	0	0	0	1	0	0
M	0	0	0	0	0	0	0	0	1	0	0	0	1	0
N	0	0	1	1	0	0	1	0	0	0	0	0	1	0

Respective result:



Excel document with matrices are here. I am not sure that 1 on diagonal should be considered as loops (most likely no, but I draw them) as well as other numbers than 1 in matrix (I had 2 for edge E-G I draw 2 edges, but it seems like in this task it doesn't matter what number, it is still only 1 edge between nodes). So in this case graph G*G shows paths of length 2 of G.

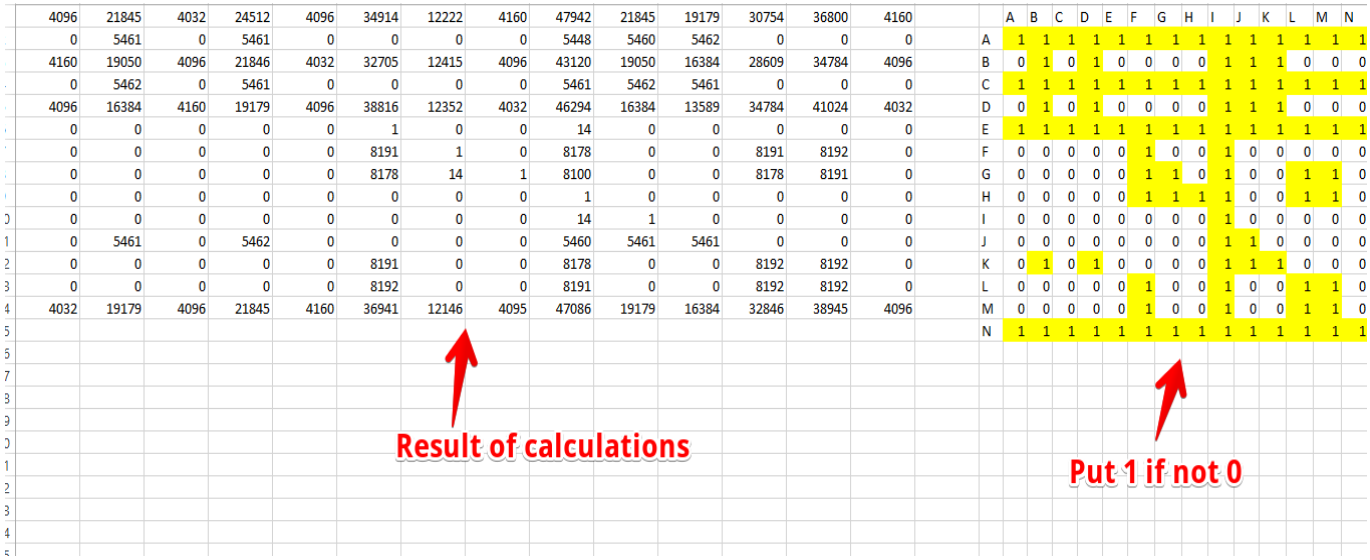
---

*2. Calculate the transitive closure of the G using the methods a) G*G*G...*G , b) G2=G*G , G4=G2*G2, ... and b) the Warshall algorithms. Verify that the methods produce the same end result.*

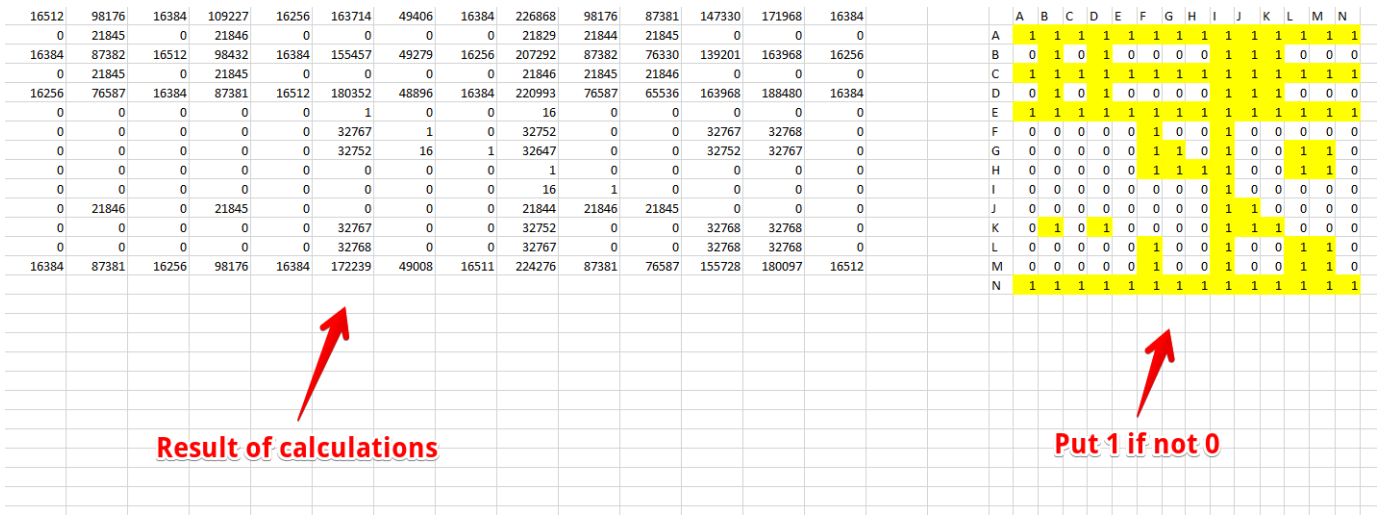
In this task I will put "1" in diagonal since we need not exactly n paths but up to n paths.

Excel file with initial matrix is here.

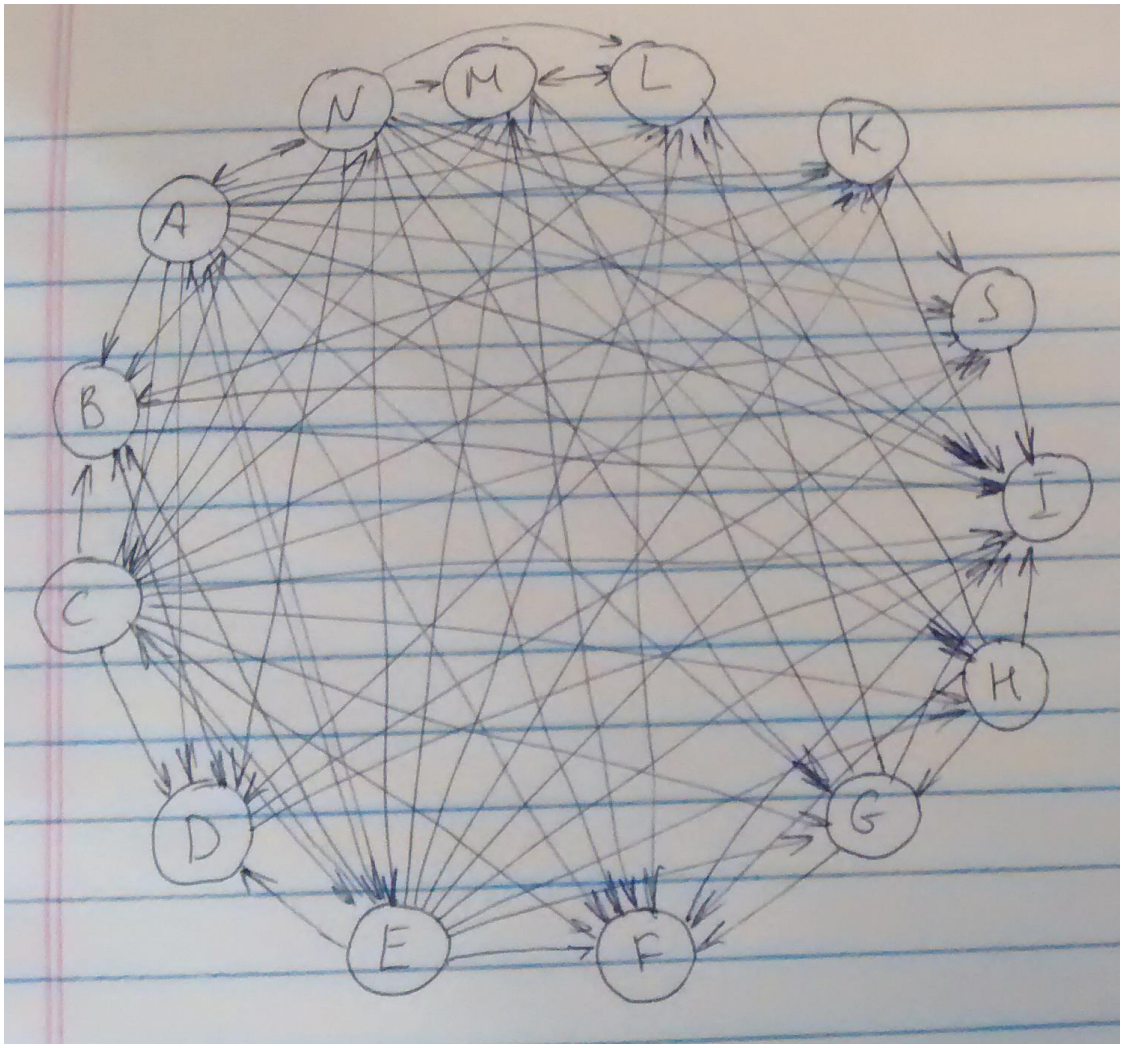
a) Calculations for both a and b available here (MathCad). Result is here tab "a":



b) Since we need to multiply previous result iteratively and there are 14 nodes in graph, basically I calculated G^16. Calculations for both a and b available here (MathCad). Result is here tab “b”:



c) Code is here. Results are the same for all 3 algorithms. Transitive closure is below:



---

3. Simulate the random walk (Markov Chain) procedure starting from A. When you have 2 edges, use 50:50 probability, when 3 then 1/3 probability, and so on. Count the nr of times you visit each node. You may observe getting stuck. Add a link from "dead end" I to A. Now repeat.

The code is here.

Since probability is equal for all nodes, I didn't use any additional conditions, just selected node randomly.

Number of times every node was visited:

{'A': 11, 'M': 10, 'D': 9, 'I': 9, 'L': 6, 'C': 5, 'B': 5, 'F': 5, 'K': 5, 'G': 4, 'J': 4, 'E': 3, 'H': 1, 'N': 1}

4. Replicate the result of 3 by adjacency matrix representation and using matrix multiplications (with and without I->A). Compare speed, how fast can you outperform random walk by matrix operations?

---

5. Study the page rank procedure without the I->A link (you can use a simplification using random walk) - allow in 20% of cases re-appear in any of the nodes of the graph (with equal probability). List the nodes with highest "page rank" (most often visited nodes, what proportion of time would be spent in each node).

I used “Dynamical system point of view” approach from here. Highest “page rank” has nodes:

- M (17%)
- D (14%)
- F and L (10% each)
- B and J (9%)

Calculations are done in MathCad and available here.

---

6. Bonus (1p) Add 5 new nodes into the graph - two ones pointing only to A, one that A points into, and 2 without any edges in or out. What would be their respective page rank probabilities?

---

2 without any edges and 2 which point only to A has zeros – last in a rank. And one that A points to – is higher in the rank than A, C, E and is equal to 1.142*10^-4. MathCad file with calculations is here.