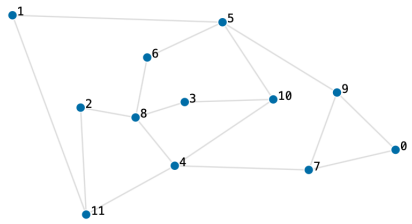
CSC 404 - Homework 4 - Name:

Problem 1. Consider the following graph



a. Using 0 as your source, construct a spanning tree for the given graph using a breadth-first process. Identify which nodes can be reached within 1, 2, 3, and 4 steps from 0.

b. Using 0 as your source, construct a spanning tree for the given graph using a depth-first process.

c. Construct the Adjacency Matrix, A (at least the first 3 rows)

$$A = \begin{pmatrix} -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

d. Determine the number of paths of length 2 that originate from 0 and end at $0, 1, 2, ..., 15$. That is, determine the first row of A^2 (count the paths by hand and use A^2 to check your work).

$$A^2 = \begin{pmatrix} -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

e. Determine the number of paths of length 3 that originate from 0 and end at $0, 1, 2, ..., 11$. That is, determine the first row of A^3 (count the paths by hand and use A^3 to check your work).

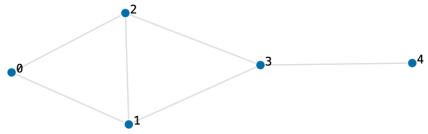
$$A^3 = \begin{pmatrix} -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

f. By using A^4, determine the number of paths of length 4 from 0 to 4. Then, list these paths :-).

Problem 2 (Let's do this with booleans - because adding is hard)**.** The following algorithm returns the Boolean product $A \odot B$ of Zero-One matrices A and B (sometimes called logical matrices).

```
1. for i = 1 to m
2.   for j = 1 to n
3.     c[i][j] = 0
4.     for k = 1 to q
5.       c[i][j] = c[i][j] ∨ (a[i][k] ∧ b[k][j])
6. return C
```

For example, for the graph G



we have

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ which gives } AA = \begin{pmatrix} 2 & 1 & 1 & 2 & 0 \\ 1 & 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \\ 2 & 1 & 1 & 3 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \text{ and } A \odot A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

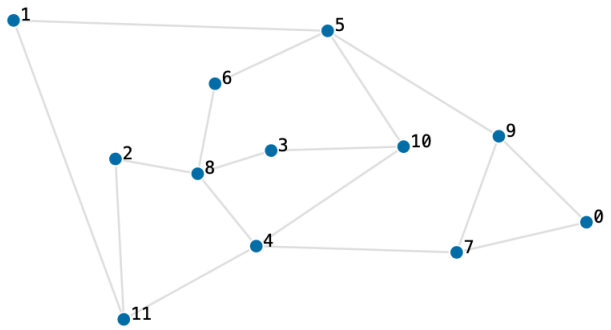
Note: For a graph with adjacency matrix A, we know the (i, j)th entry of $A^2 = AA$ gives the number of paths of length two from i to j. The (i, j)th entry of $A \odot A$ records if there exists a path from i to j (no information about the number of paths!)

The number of bit operations used to find the Boolean product of two $n \times n$ matrices can easily be determined. There are n^2 entries in $A \odot B$. Using the above algorithm, there are a total of n ORs and n ANDs are used to find an entry of $A \odot B$. Hence, $2n$ bit operations are used to find each entry. Therefore, $2n^3$ bit operations are required to compute $A \odot B$ using the above algorithm.

a. Implement the Boolean product in a language of your choice (you should be able to copy and paste Matrix Multiplication and change the operations to Boolean Operations).

b. By modifying the power matrix algorithm, we can easily create a power boolean product algorithm. Implement this algorithm in a language of your choice. (Note: We usually use A^2 to denote the boolean product $A \odot A$ – this can be dangerous since A^2 typically has other meaning, but we are lazy.)

c. Consider the following graph (Same as the last problem!):



i. By using $A^2 = A \odot A$, determine the vertices that can be reached with exactly 2 moves from 0.

ii. By using $A^3 = A \odot A^2$, determine the vertices that can be reached with exactly three moves from 0.

iii. By using $A^4 = A^2 \odot A^2$, determine the vertices that can be reached with exactly four moves from node 0.

Problem 3 (Markov and Gambling - Weee!)**.** Matching pennies is the name for a simple game used in game theory. It is played between two players, Eve (Even) and Bob (Odd). Each player has a penny and must secretly turn the penny to heads or tails. The players then reveal their choices simultaneously. If the pennies match (both heads or both tails), then Eve (Even) keeps both pennies, so wins one from Bob (Odd) i.e., +1 for Eve (Even) and −1 for Bob (Odd). If the pennies do not match (one heads and one tails) Bob(Odd) keeps both pennies, so receives one from Eve (Even), i.e., −1 for Eve (Even) and +1 for Bob (Odd).

Consider the following example, where Eve and Bob both start with two pennies. (Here the states `Si` denote how many pennies Eve has with `S0 = SL` and `S4 = SW`.

Round	Eve	Bob	Eve (pennies)	Bob (pennies)	State	Description
0	-	-	2	2	S2	Initially both have two pennies!
1	H	H	3	1	S3	Both Heads - Eve wins a Penny!
2	H	T	2	2	S2	Head and Tails - Bob wins a Penny!
3	H	T	1	3	S1	Head and Tails - Bob wins a Penny!
4	T	T	2	2	S2	Both Tails - Eve wins a Penny!
5	T	H	1	3	S1	Tails and Head - Bob wins a Penny!
6	H	T	0	4	S0 = SL	Tails and Head - Bob wins a Penny!

a. If both parties randomly flip their coin, then each have a 0.5 chance of winning. (For Eve to win we need HH or TT, so Prob(Eve Win) $= 0.5 * 0.5 + 0.5 * 0.5 = 0.25 + 0.25 = 0.5$).

i. Construct the state diagram for this game. Add loops on 1 at the two 'end game states' - `SW` and `SL` (i.e., make them absorbing states).

ii. Construct the state transition matrix, P, for this game. Label the rows SL, S1, S2, S3, SW. (You should see a 1 in the top left and bottom right)

$$P = \begin{array}{c} \\ SL \\ S1 \\ S2 \\ S3 \\ SW \end{array} \begin{array}{c} \begin{array}{ccccc} SL & S1 & S2 & S3 & SW \end{array} \\ \begin{pmatrix} 1 & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & 1 \end{pmatrix} \end{array}$$

iii. Compute P^{100} (or your favorite power of P). You should see a matrix with (essentially) zeros in the middle and all of the data on the left and right columns. From this, record the probability of Eve winning/losing. (Remember we start with 2 pennies, but you can also see the probabilities of starting with 1 or 3 pennies)

b. Eve has been tipped off that if Bob has three pennies, i.e., we are at state `S1`, then he is more likely to throw heads than tails (60% heads to 40% tails). With that, whenever Bob is up to three pennies Eve plans to always throw heads. Thus, at this state, `S1`, the probability for Eve to win is 0.6.

i. Construct the state diagram for this game. Add loops on 1 at the two 'end game states' - `SW` and `SL` (i.e., make them absorbing states).

ii. Construct the state transition matrix, P, for this game.

$$P = \begin{array}{c} \\ SL \\ S1 \\ S2 \\ S3 \\ SW \end{array} \begin{array}{c} \begin{array}{ccccc} SL & S1 & S2 & S3 & SW \end{array} \\ \begin{pmatrix} 1 & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & 1 \end{pmatrix} \end{array}$$

iii. Compute P^{100} (or your favorite power of P). You should see a matrix with (essentially) zeros in the middle and all of the data on the left and right columns. From this, record the probability of Eve winning/losing.

c. (Bonus) Ideas to switch from flipping pennies to playing rock-paper-scissor? Other variants?

Problem 4 (Markov Chains and Jail?!)**.** Smith is in jail and has 3 dollars; he can get out on bail if he has 8 dollars. (A strange set up, but let's go with it.) A guard agrees to make a series of bets with him. If Smith bets A dollars, he wins A dollars with probability 0.4 and loses A dollars with probability 0.6. Find the probability that he wins 8 dollars before losing all of his money in the following scenarios.

a. He bets 1 dollar each time (timid strategy). For example, if he has 5 dollars, with 0.4 probability he moves to 6 dollars and 0.6 probability he moves to 4 dollars.

   i. Draw the state diagram for the betting strategy. Add loops of 1 on the two 'end game states' - `W` (Win) and `L` (Lose).

   ii. Construct the state transition matrix P.

$$P = \begin{array}{c} \\ L \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ W \end{array} \begin{array}{c} \begin{array}{ccccccccc} L & 1 & 2 & 3 & 4 & 5 & 6 & 7 & W \end{array} \\ \begin{pmatrix} 1 & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & 1 \end{pmatrix} \end{array}$$

   iii. Compute P^{100} (or your favorite power of P) and determine the probability and Smith ends up with 8 dollars. (Remember: we started with \$3 – so we want to record the entry in the $(3, W)$ cell of P^{100}.)

b. He bets, each time, as much as possible but not more than necessary to bring his fortune up to 8 dollars (bold strategy). For example, if he has 5 dollars, then he will bet 3 dollars – i.e., 0.4 probability he moves to 8 dollars and 0.6 probability he moves to 2 dollars.

i. Draw the state diagram for the betting strategy. Add loops of 1 on the two 'end game states' - `W` (Win) and `L` (Lose).

ii. Construct the state transition matrix P.

$$P = \begin{array}{c} \\ L \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ W \end{array} \begin{array}{c} \begin{array}{ccccccccc} L & 1 & 2 & 3 & 4 & 5 & 6 & 7 & W \end{array} \\ \begin{pmatrix} 1 & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & 1 \end{pmatrix} \end{array}$$

iii. Compute P^{100} (or your favorite power of P) and determine the probability and Smith ends up with 8 dollars.

c. Which strategy gives Smith the better chance of getting out of jail?

d. (Bonus) Other betting strategies? Are they better or worse than these?

Problem 5 (Markov Chains and Craps!)**.** In the game of craps, we roll a pair of six-sided dice. On the first throw, if we roll a 7 or an 11, we win right away. If we roll a 2, 3, or 12, we lose right away. If we first roll a total of 4, 5, 6, 8, 9, or 10, we keep rolling the dice until we get either a 7 or the total rolled on the first throw. If we get a 7, we lose. If we roll the same total as the first throw, we win.

a. Construct the state diagram for this game. Let I denote the initial state (first roll). Add loops of 1 on the two 'end game states' - `W` (Win) and `L` (Lose). For the continue rolls, add loops on the outcome of the initial roll (with probability of 'roll again'). For example, if we first roll a 4, then the probability of winning on the next roll is 3/36 (i.e., roll a 4), the probability of losing on the next roll is 6/36 (i.e., roll a 7), and the probability of rolling again is $(36-9)/36 = 27/36$ (i.e., do not roll a 4 or 7). It may be helpful to remember the probability of rolling various sums

Sum	2	3	4	5	6	7	8	9	10	11	12
Prob	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36
Prob	0.028	0.055	0.083	0.111	0.139	0.167	0.139	0.111	0.083	0.055	0.028

Label the states `I, L, 4, 5, 6, 8, 9, 10, W`.

b. Construct the state transition matrix, P, for this game.

$$P = \begin{array}{c} \\ I \\ L \\ 4 \\ 5 \\ 6 \\ 8 \\ 9 \\ 10 \\ W \end{array} \begin{array}{c} \begin{array}{ccccccccc} I & L & 4 & 5 & 6 & 8 & 9 & 10 & W \end{array} \\ \begin{pmatrix} 0 & 4/36 & 3/36 & 4/36 & 5/36 & 5/36 & 4/36 & 3/36 & 8/36 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6/36 & 27/36 & 0 & 0 & 0 & 0 & 0 & 3/36 \\ 0 & -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

c. Compute P^{100} (or your favorite power of P). You should see a matrix with (essentially) zeros in the middle and all of the data on the left and right columns. From this, record the probability of winning at craps. (This should be the top right corner!)