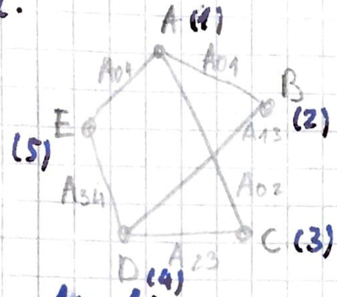
Assigment 1 Assignment 1: "Graph Theory and Networks in Python" 02.11.2021

Problem 1-1 Adjacency Matrix

$$A_G = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

1. G is undirected, as A_G is symmetric (i.e. $(A_G)_{ij} = (A_G)_{ji}$ $\forall i,j = 0,\dots,5$)

2.

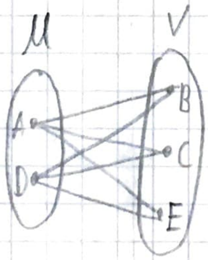


$A_{ij} = A_{ji}$

graph described by A_G

→ nodes {A, B, C, D, E} correspond to row/columns 1,2,3,4,5 of the adjacency matrix

3. yes, we can divide the nodes N into two disjoint sets U= {A, D} and V= {B, C, E} so that every link connects only a node in U to a node in V (but not to a node within the same set). This means that U and V are independent sets. An illustration can be seen here:



4. adjacency list:

node	linked to
A	B, C, E
B	A, D
C	A, D
D	B, C, E
E	A, D

edge list

pair of edges
(A, B)
(A, C)
(A, E)
(B, D)
(C, D)
(D, E)

Problem 1-2 Average Degree of a Growing Network

1. total number of nodes N:

$N(t) = t \curvearrowright N(t=T) = \underline{\underline{T}}$ (with $N(1)=1$, $N(2)=2 \dots$)

2. total number of links L:

$L(t) = t-1 \curvearrowright L(t=T) = \underline{\underline{T-1}}$ (with $N(1)=0$, $N(2)=1, \dots$)

3. average degree $\langle k \rangle$:

$\langle k \rangle(t) = \frac{2L}{N} = \frac{2(t-1)}{t} \curvearrowright \langle k \rangle(T) = \underline{\underline{\frac{2(T-1)}{T}}}$ (with $\langle k \rangle(1) = 0$, $\langle k \rangle(2) = 1$, $\vdots$)

4. average ~~limit~~ degree in the limit $T \to \infty$

$\lim_{T\to\infty} \langle k \rangle(T) = \lim_{T\to\infty} \frac{2(T-1)}{T} = \lim_{T\to\infty} 2(1-\underbrace{\tfrac{1}{T}}_{\xrightarrow[T\to\infty]{} 0})$

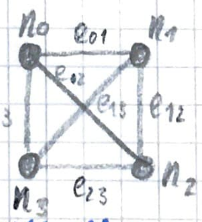
$= \underline{\underline{2}}$

Problem 1-3 Difficulty of an Exhaustive Search

Number of walks that need to be checked for an eulerian trail for:

1. complete graph with $N=4$ nodes:

sketch:



average degree $\langle k \rangle = N-\underline{1}$ ($k = \langle k \rangle$ ∀nodes $n_0, \dots, n_3$)
total number of links $L = 6 = \underline{\frac{\langle k \rangle \cdot N}{2}}$

Procedure for checking for eulerian trails as described in the problem description (exhaustive search):

(a) start at any of the nodes $n_{i'}$ (4 possibilities $=N$)

(b) traverse an edge $e_{i'} = (n_{i'}, n_i)$ ($i \neq i'$)
($k = \langle k \rangle = 3$ possibilities)

(c) repeat (b) $l = L$ times so that a walk of length $l = L$ is performed ($L=6$ times)

$\Rightarrow$ #walks checked $= N \cdot \langle k \rangle^L = 4 \cdot 3^6 = \underline{\underline{2916}}$
$= N \cdot k^L$