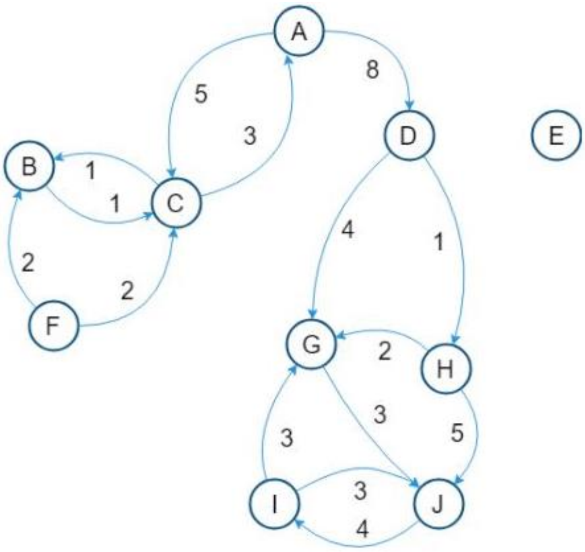
Assignment 3

Task 2.1 Adjacency Matrix

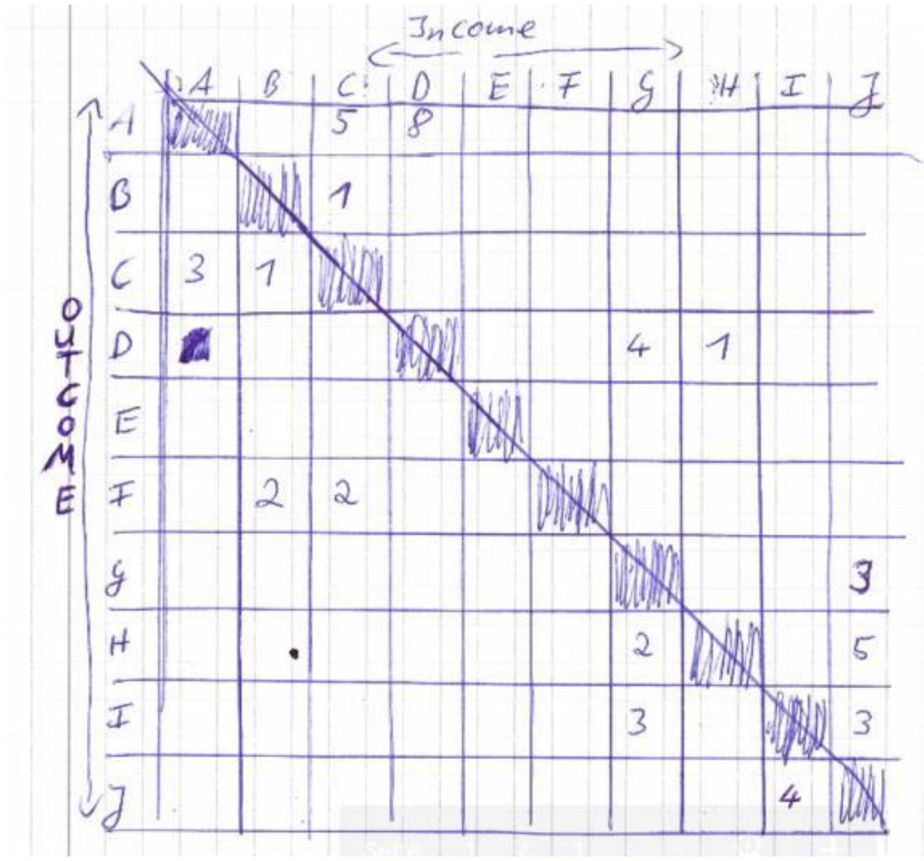
Main principles:

1. Node list as rows and columns of table
2. Cell shows presence/absence of edge
3. Edge between nodes can be weighted or have attributes
4. If the edges are directed one side represents the outcome and the other one the income
5. Takes network nodes as keys and link status between nodes as values
6. Scale: Nodes: 1000. Links: one million

Graph:



Matrix:


Income
OUTCOME

	A	B	C	D	E	F	G	H	I	J
A	▨		5	8						
B		▨	1							
C	3	1	▨							
D				▨			4	1		
E					▨					
F		2	2			▨				
G							▨			3
H							2	▨		5
I							3		▨	3
J									4	▨

Visibility:

Graph	Matrix
Cycles and Subgraphs or Connections (not the directed ones) are easy to find Example: E is not connected to any other node	If the Edges are not directed, we can see subgraphs and the way how nodes are connected If there is one big line in one row or column one node is connected to many other ones
Hierarchical structures can be easier analysed **if** the graph is structured like the given one Example: A is the root node because every subgraph can be reached from this node, but not every node can reach A	Out- and Income can be compared fast (in our case **income** columns and **outcome** rows)
• topology understanding, path tracing • intuitive, no training needed • node-link best for **small networks**	• predictability, scalability, supports reordering • some topology tasks trainable • matrix best for **large networks**