# Programming with Data – Coursework II
# Problem solving and Scalability

The goal of this coursework is to make you apply the concepts and general methods seen in class and understand the scalability of your code. This assignment requires you to upload a short Python source code and an essay/technical annex that describes i) your analysis of the problems described below, ii) your algorithmic solution and the Python code for it and iii) the scalability analysis of your code, tested against random instances of the problem. Work is organized in phases, as described below.

**Corporate Control.**
**Instance:**
-a set of companies, identified by integers over 0..n-1.
-a cross participation matrix A: a[i][j] represents the percentage of capital that company i holds[1] of company j.
-an arbitrary company s (for seed).

Conventionally, a[i,i]=0 for all companies i but at the beginning of the computation we seed the control structure with a[s][s]=1.
Hence, the main diagonal will be 0 everywhere but for s.
Also, notice that a[i, j] <> a[j, i] in general.

**Solution:**
all companies controlled by the seed company s, either directly or indirectly, with the following definition of control:
if a[i,j]>0.5 then i controls j
if i controls {c_1, … c_k} and the sum of all participations in j by i, c_1, … c_k is >0.5 then i controls j.
Example:

```
A = [[0.0, 0.6, 0.20],
     [0.1, 0.0, 0.35],
     [0.0, 0.1, 0.00]]
s = 0
```

Here company s(=0) controls company 1 directly, with 60% of the shares.
Also, it controls company 2 *indirectly*: the sum of all controlled participations in 2 is 55%.

1 This is a simplified model where we assume that all shares have exactly the same voting rights, i.e., i) holding X% of the shares gives X% of “control,” and ii) the combined contol of 51% or of the shares gives total control of a company.

Phase 1.
Design and code a Python function that generates random company participation matrices (the A matrix seen above).
Write a section of the technical annex explaining the choices you made, the parameters used by your Python function and why you consider the output matrix a realistic instance of cross-participation.
Extra question: allow a parameter to select *Markovian* instances. That is, each column of $A_M$ must exactly sum to 100%.

Phase 2.
Design an algorithm that solves Corporate control, under the assumptions seen above.
Implement the algorithm as a Python function which takes a control matrix (see Phase 1 above) and a seed company s as input and returns the list of all companies that are controlled, directly or indirectly, by s.
Test your implementation against random instances created by your function above. On small instances (n=3 or 4), is the result correct?
Write a section of your technical annex explaining your method and why you believe that it is correct (an informal argument suffices).

Phase 3.
Run scalability testing: define a set of exponent values (e.g., k=[2..8]} and for each k set $n=10^k$ and generate 10 random instances of size k (see Phase 1 above). Now, run your solution, measure the running time (see notes from Class 2) and compute the average. Plot the results and insert them in the technical annex.
Optionally: find the best-fit curve (not defined in this module) of your average times; report the results and comment them in the technical annex.
Extra question (if you implemented Markovian matrices in Phase1): are the times of your function sensitive to Markovian inputs? Please devise a comparative experiment anc comment the results accordingly.

**Important dates:** Please upload your essay to Moodle by 23:55 of Monday May 8th 2018.

**Submission**: Please upload the python source code of your solution and a technical annex.
About the technical annex: please use your judgement on the right amount of data and length of presentation for a technical description of your solution. In this instructor's opinion, two pages should suffice. Please submit your technical annex in a mainstream format (e.g., HTML, PDF, ODT, TEX, MD) that support your presentation style. Please also be advised that heavy Microsoft Office formats (e.g., DOCX, PPTX) could cause anomalies when displayed by Libre Office[2] so please refrain from using them. As a courtesy, please use an easy-to-read style similar to that of this document (Times New Roman font or similar, size 12 or bigger, 1.15 line spacing or higher, justified alignment).

2 https://www.libreoffice.org/

**Plagiarism**: please be advised that Moodle deploys a state-of-the-art plagiarism detection software[3] to evaluate coursework submissions against both Web sources and other submissions, past and present. Each submission will be scored for originality; submissions with low originality might be discarded or penalized.
It is however possible to insert quotations by using appropriate typographic style and providing the reference:

*this phrase is an example of a typographic style for citation and reference: it will be discounted by Turnitin analysis.*

Please make use of a formalized citation system and report articles and books you refer to, e.g. [Narayanan et al., 2011] and [Shenoy et al., 2011].

## References

[Shenoy et al., 2011]
G. G. Shenoy, M. A. Wagle, A. Shaikh, 2017
*Kaggle Competition: Expedia Hotel Recommendations*
https://arxiv.org/abs/1703.02915

[Narayanan et al., 2011]
A. Narayanan, E. Shi, B. I. P. Rubinstein, 2011.
*Link Prediction by De-anonymization: How We Won the Kaggle Social Network Challenge*
Proc. of the 2011 Int'l Joint Conference on Neural Networks.
https://arxiv.org/abs/1102.4374

3 https://turnitin.com/gateway/index.html