# AG44 – Project 1 – Ariadne thread

Cédric PAQUET – Groupe A

1 ) We need to find the strongly connected components in the graph generate from the adjacency matrix A. To perform this task we have differents algorithms based on DFS. I chosed to implement Tarjan because it only require one depth-first search.

4) For the longest path I chosed to implement a variation of the famous Dijkstra algorithm but instead of choose the minimum in cost, we choose the maximum.

Data structures :

- **Node** :
  - **ID** : Unique number identifying the *node*
  - **SuccessorsList** : List of integer which contain the *ID*' of the *successors*
- **Graph**
  - **Nodes** : HashMap<Integer,*Node*> Containing every *node* of a given *Graph* referenced by their *ID*
  - **Lenght** : integer corresponding to the number of *nodes* in a given *Graph*

Input A matrix :

```
0 1 0 0 0 0 0 0
0 0 1 1 0 0 0 0
0 1 0 0 1 0 1 0
0 0 0 0 1 0 0 1
0 0 0 1 0 0 0 1
0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0
```

Output N matrix :

```
0 0 0 0
2 0 0 0
2 2 0 0
0 0 1 0
```

```
      With the groups :
```

- `Groupe 1 : 8`
- `Groupe 2 : 4 5`
- `Groupe 3 : 6 7 3 2`
- `Groupe 4 : 1`