Magnus Munch Nielsen Sørensen
mmns@ruc.dk

Documentation

Part 1 & 2 :

These first two parts are answered through the SubjectGraph class. It contains:

- SubjectGraph
  In a sense a contructor – but does also contain code that simply needs to be run.
  It maps each subject to an index in order to be able to make a hashmap, and then calls the fillMatrix method.
- set <String> gatherSubjects
  Creates a hashset containing the elements of the combi.txt file split.
- void fillMatrix
  Creates a matrix and fills like an adjecency matrix with the number of students as elements.
- void printMatrix
- boolean isConnected
  Visits every vertex to check if there is a connection to another vertex
- void recursiveFindMatch
  Is being used by isConnected. Recursively uses the DFS approch to searching vertexes.
- void printConnected
  Prints wether the the matrix is connected.

Part 3:

This part is answered partly by the NonTakenCourses class. I had issues understanding what I had to do. What it does, is that it finds all the combinations of subjectmodules that does not have an edge over 0 and therefore no connection. It then makes a list of all of them.

It contains these methods:

- List<String> findAllNonTakenCourses
  Identifies the all the edges (combinations) in the matrix where there are no students and adds them to the list.
- void printNonTakenCourses