ISLAMIC UNIVERSITY OF TECHNOLOGY

Organization of Islamic Cooperation

Board Bazar, Gazipur

Lab Report 3

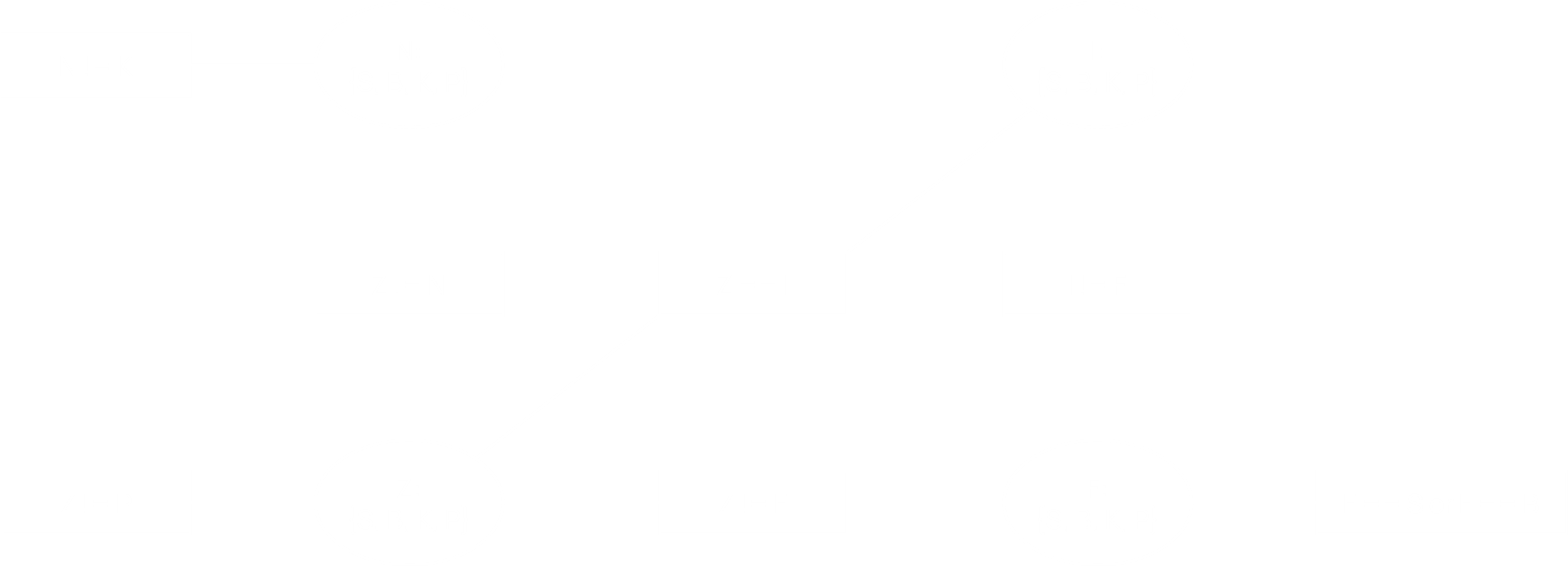
CSE 4712

Md. Bakhtiar Hasan

29th October, 2022

## Question 01

The first problem required us to create a CSP for a fairly basic problem with explicitly specified constraints. The graph created for this problem looks like this:



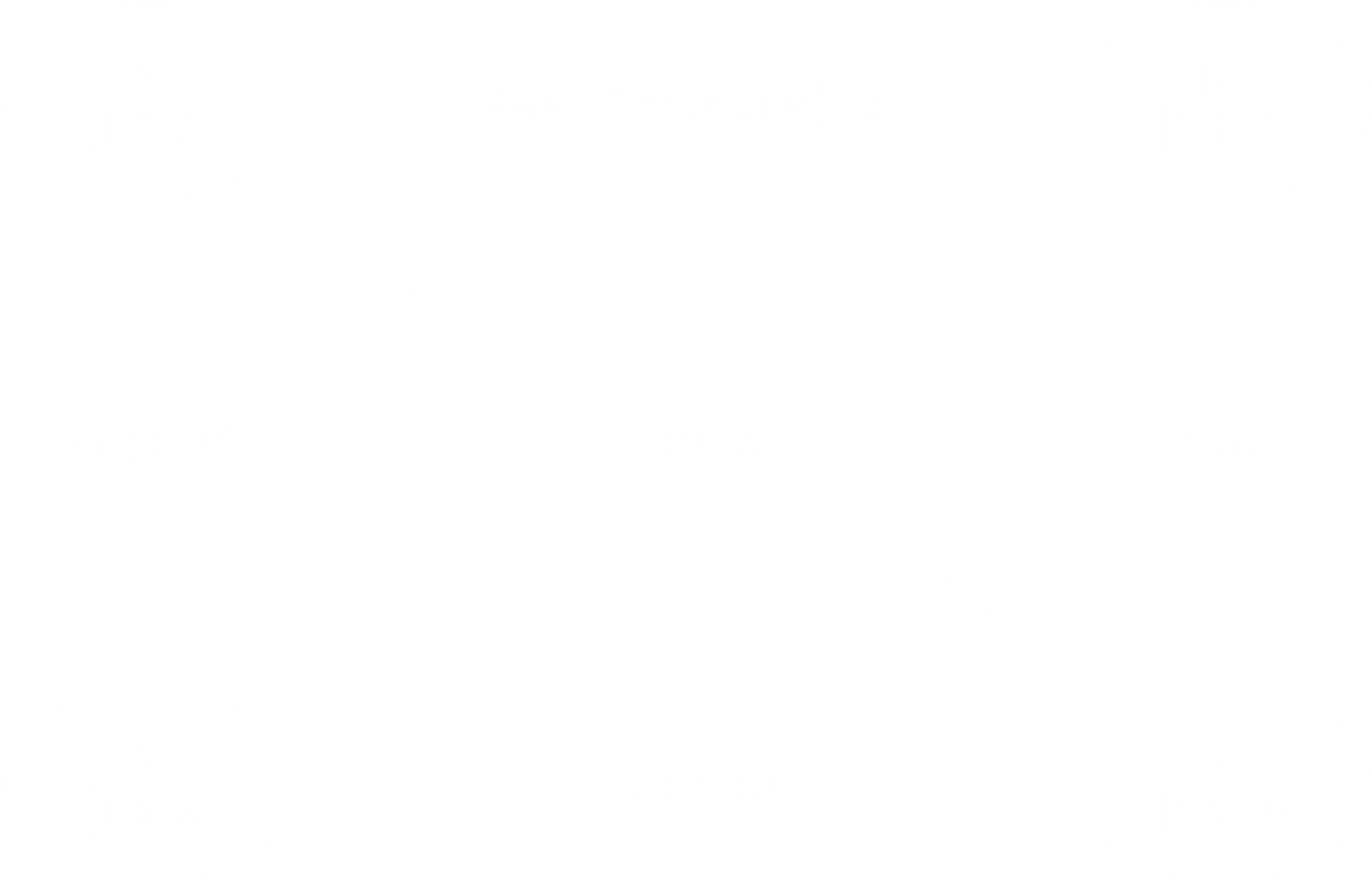
Each oval represents a variable along with its possible values. Each rectangle represents a constraint. For example, the first constraint states that .

Using the AutoSolve feature of the software on this graph gives the following solution. Note that this is just one of multiple possible solutions.

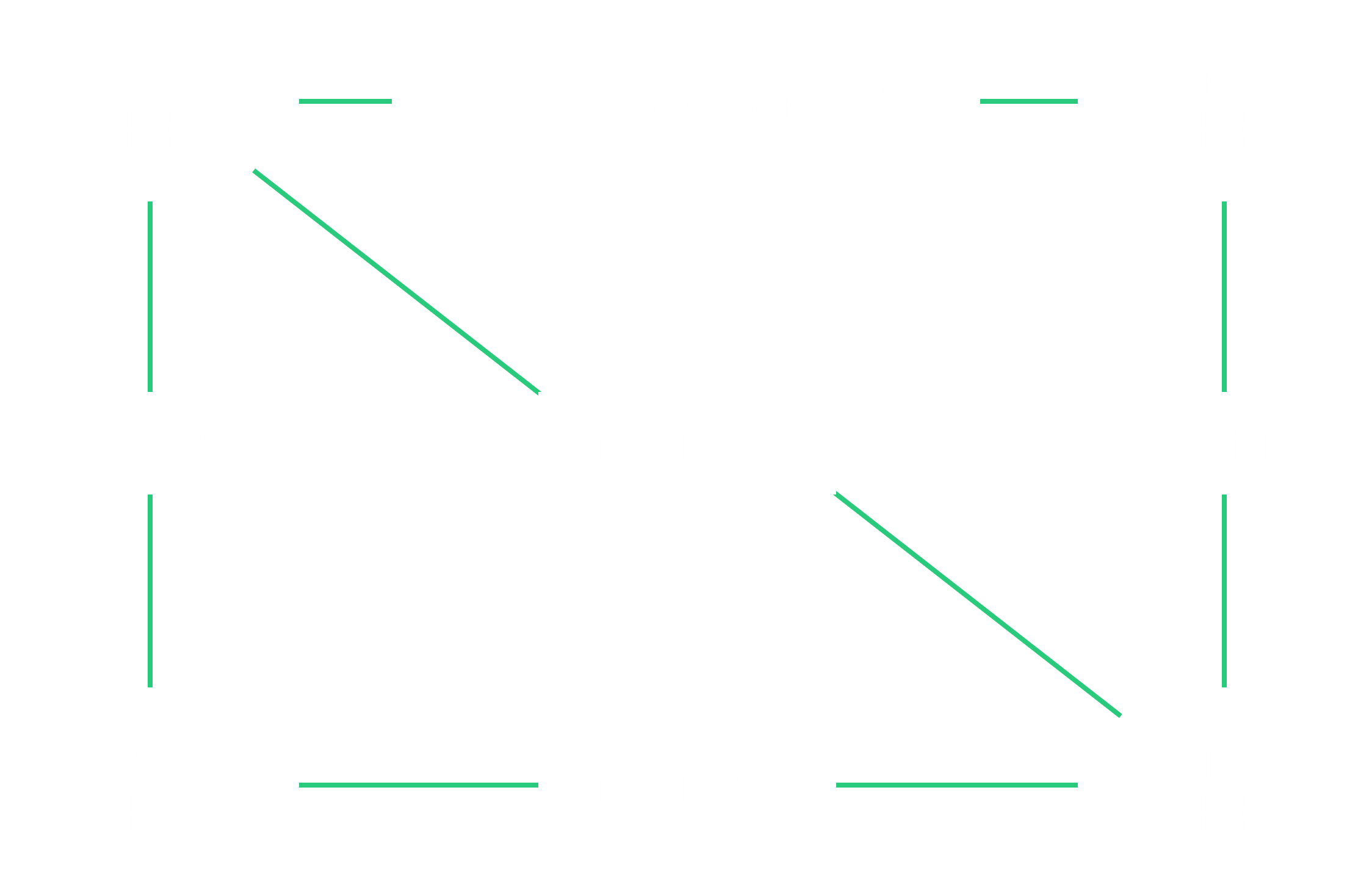


## Question 02

The second problem is similar to the first one in that it also has explicitly specified constraint. Thus, it is possible to easily create a graph for this problem.

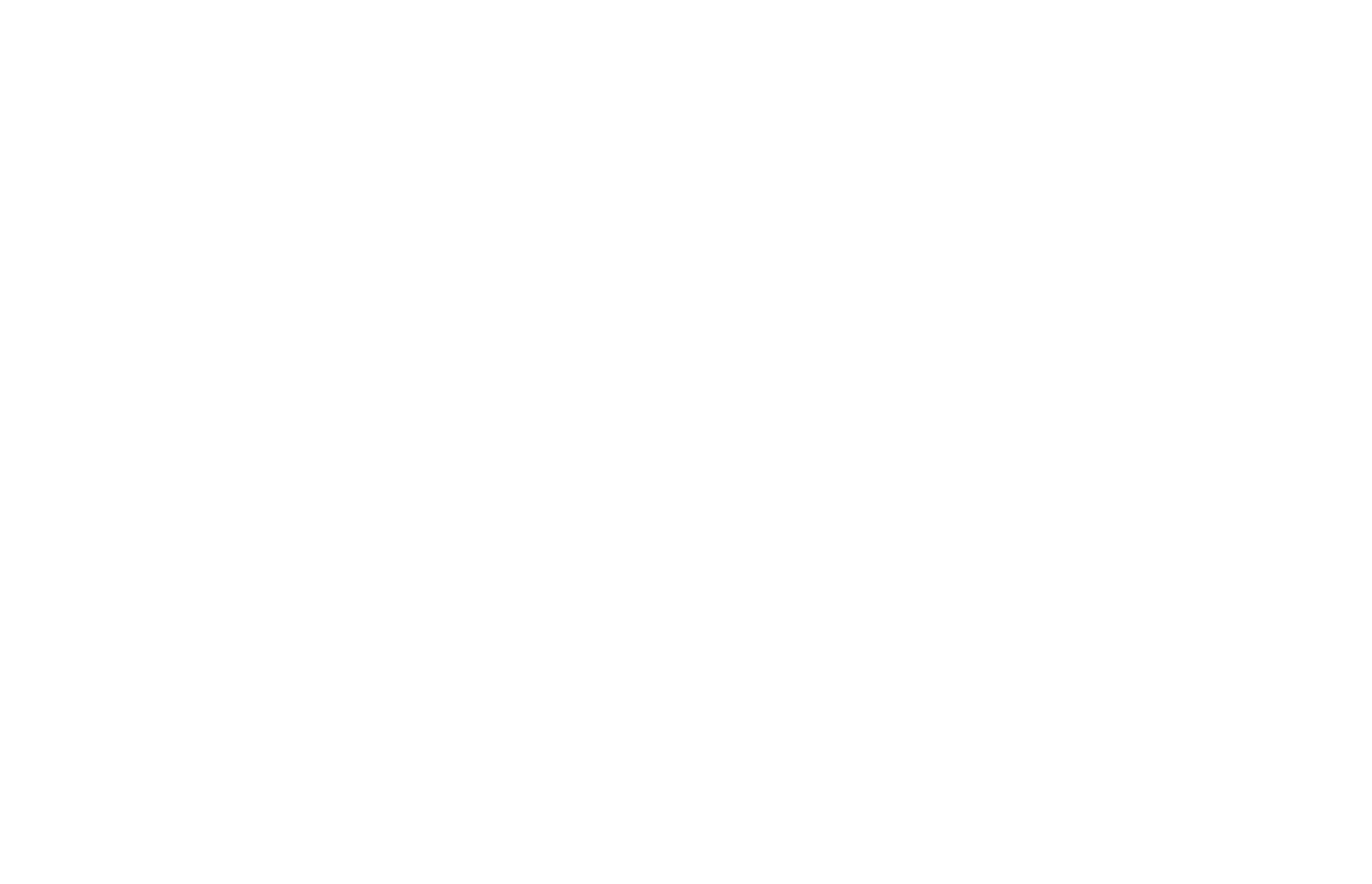


Using the AutoSolve feature on this graph gives us the following result:



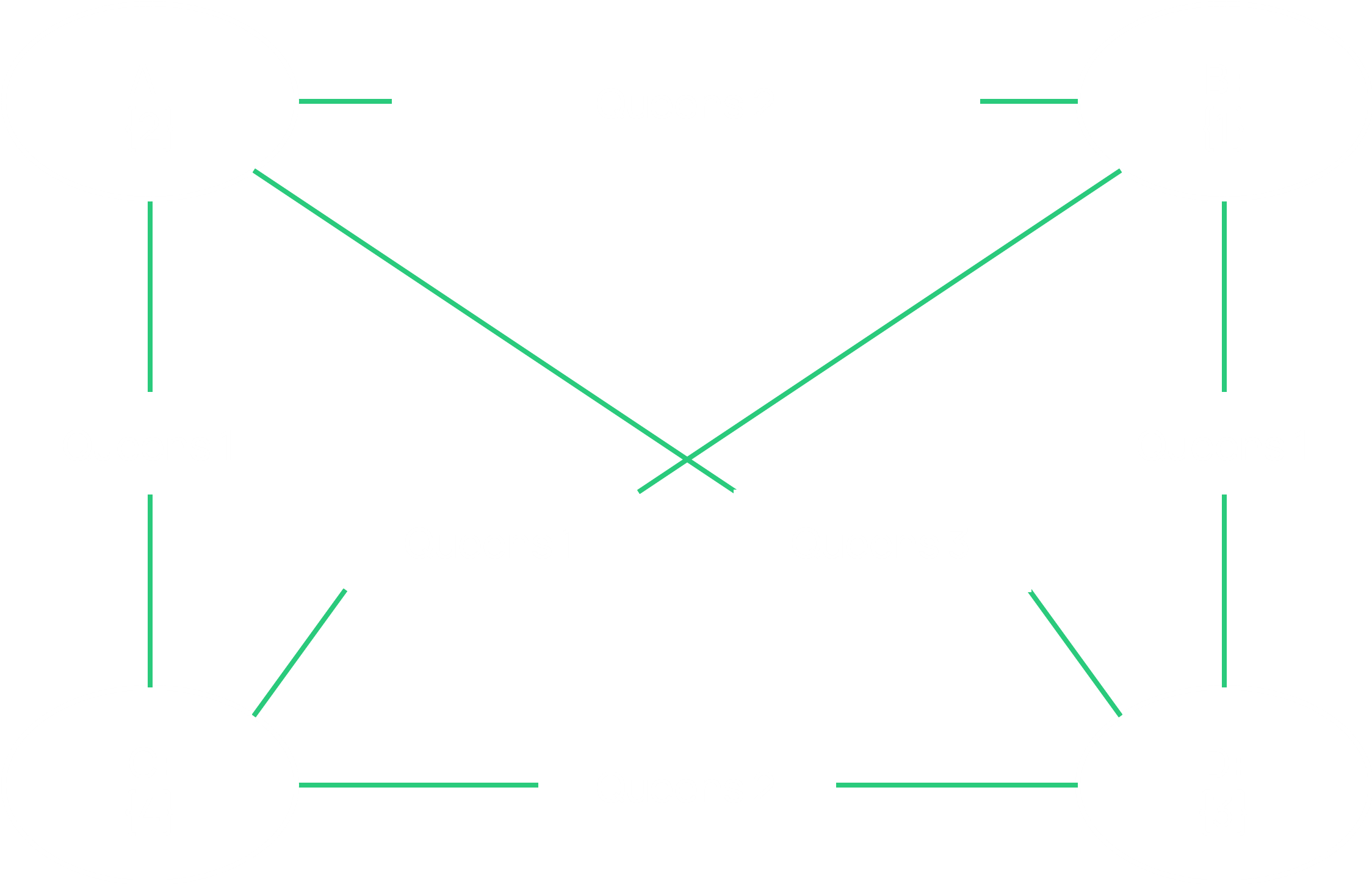
## Question 03

The third problem is the 4 Queens problem. The problem requires us to find the position of 4 queens, each on a different row, such that they are all unable to attack each other. The graph for the problem looks like this:



The constraints for this problem are quite complicated, since we need to ensure two queens are never on the same row, column or diagonal. Setting this up manually would be very difficult. However, the software provides an easy workaround for this by using a built-in constraint type created specifically for the N-Queens problem.

The solution using AutoSolve is as follows:



## Question 04

The fourth problem was comparatively far more difficult than the previous ones. This is because it is essentially two CSPs combined. We were required to find the times at which different tasks would be completed, i.e., set up a schedule, but also find which of two faculty members would oversee the tasks. The number and complexity of the constraints involved made this problem exceptionally difficult.

The proper solution to this problem seems to be a two-part CSP, with the first part dealing with scheduling faculty members to time slots such that they do not overlap, and the second part dealing with scheduling classes to time slots. Unfortunately, a clean solution to this problem could not be modelled, so a roundabout approach is being taken.



In the graph above, each class is considered a variable. The domains for the variables are tuples of starting times and faculty members, meaning there are 8 possible values. The constraints are all created manually. The constraint names as seen in the graph are written in shorthand, with the subscript indicating whether the constraint is concerned with the faculty member, the time or both. For example, refers to the fact that both faculty members must be present when the TRW Lab is being taken, while refers to the fact that if some faculty member is taking the AI Lab at a specific time, the same faculty member cannot be checking quiz scripts at that time.

Using the AutoSolve feature, the solution to the above graph is as follows:

