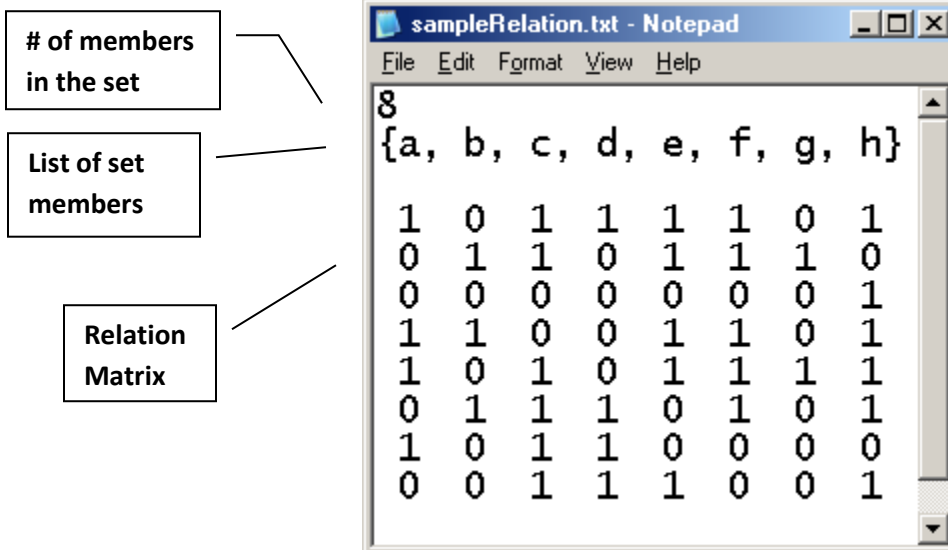


Programming Assignment for ICS-254

Relations Analyzer

Write a program that does the following:

- (a) Takes as input a finite relation in matrix form. You could use the following notepad file as a guideline.



- (b) Your program then does the following processing on this relation

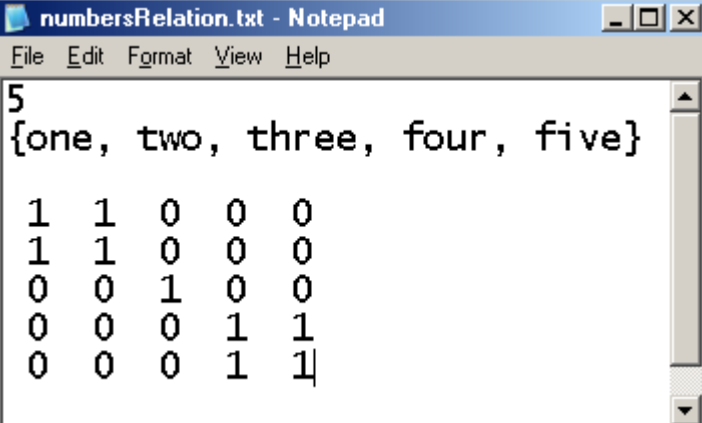
BASIC FUNCTIONALITY (Phase 1: 2%): The program finds and states the following properties for the input relation.

- (i) reflexivity (reflexive, irreflexive or none)
- (ii) symmetry (symmetry, asymmetry, antisymmetry or none)
- (iii) transitivity (transitive or not)
- (iv) whether the input relation is an equivalence relation or a partial ordering.

A sample output could be:

The input relation is reflexive, antisymmetric and transitive. Hence it is a partial ordering.

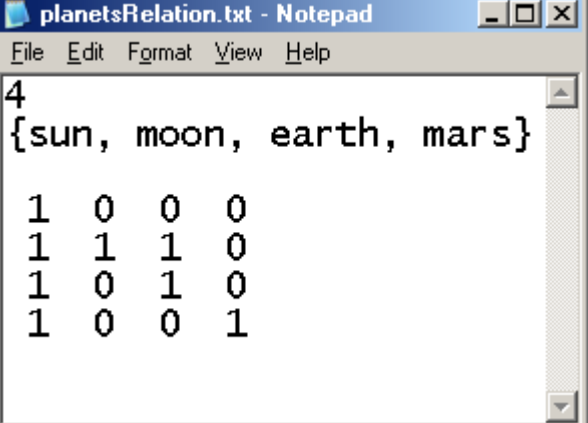
INTERMEDIATE FUNCTIONALITY (Phase 2: 3%): If the relation is an equivalence relation, the program finds and states all the equivalence classes in it. For example the following relation is an equivalence relation:



The equivalence classes in this relation are

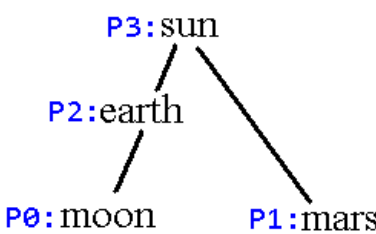
EQ1: {one, two}
EQ2: {three}
EQ3: {four, five}

If the relation is a partial ordering, the program finds and states all the elements in its Hasse Diagram. For example, the following relation is a partial ordering (Hasse Diagram also shown). Also find and print the maximal, minimal, greatest and the least elements.



The elements in the Hasse Diagram can be stated as follows:

P0: (-, {moon})
P1: (-, {mars})
P2: (P0, {earth})
P3: (P1 P2, {sun})



```

graph BT
    P0["P0:moon"] --> P2["P2:earth"]
    P1["P1:mars"] --> P2
    P2 --> P3["P3:sun"]
  
```

Submission Guidelines

1. This assignment is worth 5% of the grade.
2. This assignment is supposed to be done individually.
3. Use the java programming language for this assignment.
4. Submit your java files (as a zip archive) along with a small **readme.txt** file, showing the usage of your program.