Report for discrete maths lab work

For more detailed documentation, check the code's documentation

For the lab work, we used python, which is counterproductive.

The functions are as such:

- symmetric_closure - makes a symmetric closure of matrix representation of a relation
- symmetric_check - check if the matrix representation is symmetric
- reflexive_closure - makes a reflexive closure of matrix representation of a relation
- reflexive_check - check if the matrix representation if reflexive
- transitive_closure - makes a transitive closure of matrix representation of a relation
- transitive_check - check if a relation is transitive
- equivalence_class - creates a davision of a set by the equivalence relation (if makes it equivalent)

The next functions are rather self-explanatory

- read_file
- write_matrix
- write_set

Here's the breakdown of all the complexities:

- symmetric_closure - $O(n^2)$
- symmetric_check - $O(n^2)$
- reflexive_closure - $O(n)$
- reflexive_check - $O(n)$
- transitive_closure - $O(n^3)$
- transitive_check - $O(n^3)$
- equivalence_class - $O(n^2)$
- read_file - $O(n^2)$
- write_matrix - $O(n^2)$
- write_set - $O(n^2)$

Notes

- **There is a check of matrixe's squareness in all the closures**

```
assert all(len(matrix) == len(x) for x in matrix) and matrix is not None
```

- **Also, the transitive check is O(n^3) because the function makes a transitive closure of a matrix**