Problem 1.

What is the worst-case complexity of computing the inverse relation in case where the relation is represented

a) by the set of pairs of the relation;

b) by the matrix of the relation?

Solution: Let $R \subseteq A \times A$ is the binary relation.

Definition

$R^d := \{(a, b) \mid (b, a) \in R\}$

a) Let S is the set of pairs for relation R. Then to compute inverse relation we need to iterate through every pair in the set S and push to the new set S_1 pairs from the first one set S with swapped elements (for example, by the algo $std :: swap$ complexity of which one is $\mathcal{O}(1)$). Then we need at least $\mathcal{O}(n)$ operations, where n is an amount of pairs in the set S.

b) Let's define the matrix of the relation by notation M_R. From the definition of the inverse relation we can obtain that our goal is to find transposed matrix to M. The worst-case time complexity to do it is $\mathcal{O}(n \times m)$, where n, m are the matrix shapes, because we need to iterate through matrix by two cycles.

■

Problem 2.

What is the worst-case complexity of computing the product of relations in case where the relation is represented

a) by the set of pairs of the relation;

b) by the matrix of the relation?

Solution:

Definition

$P \cdot R = \{(x, y) \mid \exists z \ \ ((x, z) \in P \text{ and } (z, y) \in R)\}$, where $P \subseteq A \times A$, $R \subseteq A \times A$ are two binary relations.

a) Let S_1 be the set of pairs for P and S_2 same for R. To find the product of relations we need to iterate though all pairs $(a, b) \in S_1$ and for each such pair we need to iterate through all the pairs of a kind $(b, c) \in S_2$ and push it to final set with result of the product. For such operations we need $\mathcal{O}(n \times m)$, where n is the size of set S_1, m is the size of set S_2

b) Lets define M_P and M_R as a matrices for P and R relations. The result matrix element i, j can be calculated by the formula:

$$\bigcup_{k=1}^{n} x_{ik} \wedge y_{kj}$$

We need to process all elements from $P[i, :]$ and $R[:, j]$. So there are n^2 elements in a result matrix M, so we have a worst-time complexity of $\mathcal{O}(n^3)$.

■

Problem 3.

What is the worst-case complexity of computing the transitive closure of a relation in case where the relation is represented

a) **by the set of pairs of the relation;**

b) **by the matrix of the relation?**

Solution:

Definition

$$R^T = R \cup R^2 \cup R^3 \cup \ldots = \bigcup_{i=1}^{\infty} R^i,$$

where

$$R^n = \underbrace{R \times R \times \ldots \times R}_{n \text{ times}}$$

a) We need to compute all the R^i up to $n-1$, where n is the power of set of pairs, using algorithm from the previous problem. So in the worst case we have $\mathcal{O}(n^3)$ complexity.

b) We can view on the matrix of the relation as an adjacency matrix of some graph G. Then the complexity of finding transitive closure will be the complexity for breadth-first search $\mathcal{O}(V+E)$

■