

1. Relacje

1 Uwagi ogólne

Celem ćwiczenia jest zapoznanie się z operacjami na strukturach i tablicach struktur.

Oprócz zadań opisanych w kolejnych sekcjach, program należy dodatkowo uzupełnić o funkcje:

1. `add_to_relation()`, która dodaje parę liczb do relacji, jeżeli ta para jeszcze w tej relacji nie występuje,
2. `read_relation()`, która czyta liczbę par relacji (n), a następnie n par liczb całkowitych definiujących relację,
3. `print_int_array()`, która wypisuje długość tablicy całkowitej (n), a następnie n wartości tej tablicy.

2 Zadania

2.1 Własności relacji

Szablon programu należy uzupełnić o definicję funkcji:

1. `is_reflexive()`, która sprawdza, czy relacja jest zwrotna
2. `is_irreflexive()`, która sprawdza, czy relacja jest przeciwwrotna
3. `is_symmetric()`, która sprawdza, czy relacja jest symetryczna
4. `is_antisymmetric()`, która sprawdza, czy relacja jest antysymetryczna
5. `is_asymmetric()`, która sprawdza, czy relacja jest asymetryczna
6. `is_transitive()`, która sprawdza, czy relacja jest przechodnia

Użyteczny link:

https://en.wikipedia.org/wiki/Homogeneous_relation.

- **Wejście**
1 n (liczba elementów relacji)
 n linii składających się z par liczb naturalnych (`first`, `second`)
- **Wyjście**
Sześć wartości 0 lub 1 reprezentujących własności zadanej relacji
- **Przykład:**
Wejście:

```
1 10
2 3
2 4
3 4
0 1
0 2
0 3
0 4
1 3
1 2
1 4
```

Wyjście:

```
0 1 0 1 1 1
```

2.2 Porządki, wartości minimalne i maksymalne

Szablon programu należy uzupełnić o definicję funkcji:

1. `is_partial_order()`, która sprawdza, czy relacja jest relacją częściowego porządku
2. `is_total_order()`, która sprawdza, czy relacja jest relacją całkowitego porządku
3. `find_max_elements()`, która dla relacji porządku znajduje elementy maksymalne
4. `find_min_elements()`, która dla relacji porządku znajduje elementy minimalne
5. `find_domain()`, która znajduje dziedzinę relacji R (zbiór X , na iloczynie kartezjańskim którego opisana jest relacja)

Relacja $R \subseteq X \times X$ jest relacją częściowego porządku jeżeli jest zwrotna, antysymetryczna i przechodnia.

Element $g \in X$ jest elementem maksymalnym jeżeli nie istnieje element $x \in X$, taki że $x \neq g$ i gRx . Podobnie element $m \in X$ jest elementem minimalnym jeżeli nie istnieje element $x \in X$ taki, że $x \neq m$ i xRm .

Zbiór X częściowo uporządkowany przez relację R może zawierać kilka elementów maksymalnych / minimalnych.

Relacja $R \subseteq X \times X$ jest spójna jeżeli dla każdych $x, y \in X$ zachodzi xRy lub yRx (lub jedno i drugie).

Jeżeli relacja $R \subseteq X \times X$ jest relacją częściowego porządku oraz jest spójna, to zbiór X jest liniowo (całkowicie) uporządkowany przez R .

Dziedzinę (zbiór X) wyznaczamy jako unikalną tablicę poprzedników i następników par należących do relacji.

Pomocne linki:

1. https://en.wikipedia.org/wiki/Partially_ordered_set.
2. https://en.wikipedia.org/wiki/Total_order
3. <https://www.bookofproofs.org/branches/total-order/>

- **Wejście**

2 n (liczba elementów relacji)

n linii składających się z par liczb naturalnych (**first**, **second**)

- **Wyjście**

Dwie wartości 1 lub 0 (relacja jest / nie jest relacją częściowego / całkowitego porządku)

d (liczność domeny relacji)

d liczb całkowitych (domena relacji)

max (liczba wartości maksymalnych)

max liczb całkowitych (wartości maksymalne)

min (liczba wartości minimalnych)

min liczb całkowitych (wartości minimalne)

Uwaga: Wszystkie trzy tablice wyjściowe powinny zawierać unikalne liczby całkowite w porządku rosnącym.

- **Przykład:**

Wejście:

```
2
12
1 4
1 1
1 5
1 6
2 4
2 2
2 6
3 4
3 3
4 4
6 6
5 5
```

Wyjście:

```
1 0
6
1 2 3 4 5 6
3
4 5 6
3
1 2 3
```

2.3 Złożenie relacji

Szablon programu należy uzupełnić o definicję funkcji `composition()`, która wyznacza złożenie dwóch zadanych relacji.

Użyteczny link:

https://en.wikipedia.org/wiki/Composition_of_relations

- **Wejście**

3

n1

relacja R – n1 par liczb całkowitych

n2

relacja S – n2 par liczb całkowitych

- **Wyjście**

n3 – liczba elementów relacji złożonej R o S

- **Przykład:**

Wejście:

3

7

1 2

2 3

3 4

3 2

2 5

1 5

2 4

6

2 4

1 3

5 4

3 5

3 1

1 2

Wyjście:

5