Лабораторная работа № 2
Топологическая сортировка

Краткий теоретический материал

Строгим частичным порядком на множестве X называем бинарное отношение $\prec$ на X, удовлетворяющее свойствам:

- $x \nprec x$ (антирефлексивность);
- $x \prec y$ и $y \prec z \Rightarrow x \prec z$ (транзитивность).

Элемент x частично упорядоченного множества X называется *минимальным*, если в множестве X нет элементов, меньших, чем элемент x.

Мы будем предполагать, что X – конечное множество.

Задача *топологической сортировки* состоит в том, чтобы «перевести частичное упорядочение в линейное упорядочение», т. е. расположить элементы в такую последовательность $x_1, x_2, \dots, x_n$, что если $x_j \prec x_k$, то $j < k$.

Пример. Рассмотрим множество $X = a_1, a_2, a_3, a_4, a_5$, отношение частичного порядка на котором задаётся неравенствами: $a_1 \prec a_2, a_3 \prec a_2$, $a_2 \prec a_4, a_2 \prec a_5$.

Один из способов линейного упорядочивания этого множества выглядит так: $\mathrm{a}_1, a_3, a_2, a_4, a_5$.

Псевдокод алгоритма топологической сортировки:

Алгоритм 1.12. Алгоритм топологической сортировки

Вход: конечное частично упорядоченное множество U.
Выход: линейно упорядоченное множество W.

while $U \neq \varnothing$ **do**
  $m := \min(U)$ { функция min возвращает минимальный элемент }
  yield m { включаем элемент m в множество W }
  $U := U - m$ { элемент m более не рассматривается }
end while

На каждом шаге этого алгоритма из множества выбирается минимальный элемент и добавляется в последовательность. Частично упорядоченное множество может содержать несколько минимальных элементов, поэтому существуют разные способы его линейного упорядочивания.

Пример. Рассмотрим работу этого алгоритма на примере приведённого выше частично упорядоченного множества. Сформируем матрицу A порядка $|X|$ следующим образом. Каждому заданному неравенству $a_i \prec a_j$ поставим в соответствие $A[i,j] = 1$, все остальные элементы матрицы сделаем нулевыми. В приведённом выше примере матрица A будет иметь вид:

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Минимальным элементам множества будут соответствовать столбцы, не содержащие единиц. В нашем примере такими столбцами будут первый и третий столбцы. Поэтому в качестве первого элемента можно взять a_1 или a_3. Пусть таким элементом будет a_1. Этот элемент надо удалить из множества. Соответственно из матрицы A надо удалить первую строку и первый столбец. Получим матрицу следующего вида:

$$\mathrm{A} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Видно, что в оставшемся множестве есть только один минимальный элемент – элемент a_3. Добавляем этот элемент в последовательность, удаляем из множества X и т. д.

Замечание. Если на каком-то шаге алгоритма оказалось, что в матрице отсутствуют нулевые столбцы, это свидетельствует о некорректном задании исходного бинарного отношения. Данное отношение не является отношением строгого частичного порядка. При реализации алгоритма такие ситуации необходимо отслеживать.

Задание

Написать программу, выполняющую дополнение частичного порядка на множестве $1,2,\ldots,n$ до линейного порядка. На вход программа получает следующие данные:

- количество элементов в множестве n;
- набор пар «a b», соответствующих неравенствам $a \prec b$.

На выходе программа должна выдавать список элементов, соответствующий полученному линейному порядку (элементы должны быть расставлены «по возрастанию»).

Программа должна обрабатывать ситуацию отсутствия минимального элемента в множестве – выдавать сообщение о том, что строгий частичный порядок на множестве задан некорректно.

Требования к отчету

Отчет по лабораторной работе должен включать:

1. Титульный лист; задание; исходный код.
2. Примеры работы программы (скриншоты).
3. Выводы.