Необходимые сведения.

**Простой граф** G(V, E) есть совокупность двух множеств – непустого множества вершин {V} и множества ребер {E} - пар различных элементов (пары неупорядочены для неориентированных графов и упорядочены для ориентированных графов).

Примеры:

1. неориентированный граф с вершинами {a, b, c, d} и ребрами  
   {(a,b), (a,c), (c,d)};
2. ориентированный граф (орграф) с вершинами {1, 2, 3, 4 ,5} и ребрами (у ориентированного графа ребра часто называют дугами) {(1,2), (2,1), (1,5), (5,3)}.

Если есть ребра с совпадающими началом и концом (петля), то граф будет с петлями. Если есть совпадающие, т.е. параллельные ребра, то граф называется мультиграф или псевдограф. Ни те, ни другие здесь не рассматриваются.

**Матрица смежности** A - один из способов хранения графа. Размер матрицы |V|x|V|, где |V| - количество вершин в графе, а значение записи Aij равно 1 или 0, в зависимости от того, существует ли ребро от вершины i до вершины j.

Матрицы смежности для вышеприведенных примеров:

1. 0 1 1 0

1 0 0 0

1 0 0 1

0 0 1 0 (3 ребра 6 единиц)

1. 0 1 0 0 1

1 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 1 0 0 ( 4 ребра(дуги) 4 единицы)

Матрица смежности неориентированного графа всегда симметрична. На диагонали матрицы смежности простого графа стоят 0.

**Матрица достижимости** R(G) [простого](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%BE%D0%B2_%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D0%B8_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2#%D0%9F) [ориентированного графа](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84) G содержит информацию о существовании путей между вершинами орграфа. Например для ориентированного графа из второго примера матрица достижимости имеет вид:

1 1 1 0 1

1 1 1 0 1

0 0 1 0 0

0 0 0 1 0

0 0 1 0 1

на диагонали всегда 1, т.е. вершина всегда достижима из себя. Значение записи R(G)ij равно 1 или 0, в зависимости от того, существует ли путь от вершины i до вершины j.

Получить R(G) достаточно просто:

R(G) = E ∨ A ∨ A2 ∨ … ∨ A|V|-1

Если вместо логических использовать арифметические операции R(G) = E + A + A2 + … + A|V|-1

то получим количество различных путей из i в j длиной не более |V|-1.

Таким образом необходимо уметь перемножать матрицы и складывать их.

**Разреженные матрицы** матрица с небольшим количеством ненулевых элементов. Например матрица размером 106 х 106 содержащая 107ненулевых значений может считаться разреженной.

Сжатое хранение строкой (CRS - compressed row storage) - один из многих способов хранения разреженных матриц. Рассмотрим матрицу смежности A орграфа G из второго примера.

0 1 0 0 1 строка с индексом[0]

1 0 0 0 0 строка с индексом[1]

0 0 0 0 0 строка с индексом[2]

0 0 0 0 0 строка с индексом[3]

0 0 1 0 0 строка с индексом[4]

Ее запись в формате CRS выглядит так:

массив значений: 1 1 1 1

массив индексов столбцов: 1 4 0 2

(их длины всегда совпадают с количеством ненулевых элементов)

массив индексации строк (количество накопленных ненулевых элементов по строкам)

0 2 3 3 3 4 - всегда на 1 больше количества строк, последнее значение равно количеству ненулевых элементов, разница между k и k+1 значением равна количеству ненулевых элементов в k строке, индексы столбцов которых указаны в массиве индексов столбцов. Запишем еще R(G) второго примера в формате CRS.

1 1 1 0 1

1 1 1 0 1

0 0 1 0 0

0 0 0 1 0

0 0 1 0 1

=

1 1 1 1 1 1 1 1 1 1 1 1

0 1 2 4 0 1 2 4 2 3 2 4

0, 4, 8, 9, 10, 12

(увеличенные промежутки разделяют разные строки для наглядности)

Необходимо разработать программу, которая предоставляет удобный функционал для работы с разреженными орграфами. На языке С++ требуется написать следующие функции:

Ввода орграфа из файла и с клавиатуры;

Вывода матрицы смежности в формате CRS в файл и на экран;

Суммирования двух матриц в формате CRS;

Умножения двух матриц в формате CRS;

Вычисления матрицы достижимости в формате CRS;

Вывода матрицы достижимости в формате CRS в файл и на экран.

Варианты по N - номеру в группе.

Количество вершин графа: 10 \* (10 + N), количество дуг:

5 \* ( 10 + N)

Все массивы - динамические, STL не использовать.