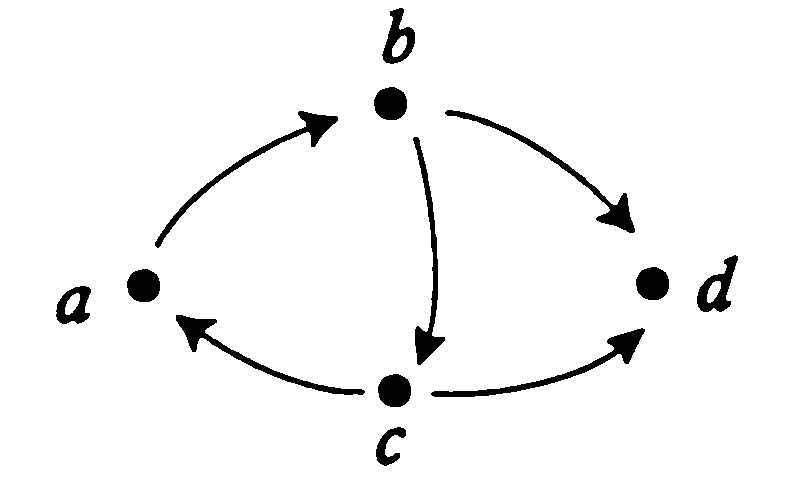
***Алгоритм Уоршелла.***

Пусть на множестве *A = {a, b, c, d}* *орграфом* и *матрицей* задано отношение *R. Булева* матрица отношения является *матрицей смежности* для орграфа.



*Рис.1.*

Построим *матрицу смежности* для нашего орграфа.

В *матрице смежности* *1* (*true*) на пересечении *i-той* строки и j-гo столбца соответствует наличию дуги, идущей от вершины с номером i к вершине с номером j. Дуга, по определению, является путем длины *1*.

Таким образом, матрица смежности указывает все пути длины *1*. В нашем случае она залает отношение *R* имеет вид:

*Булево произведение* матрицы на саму себя (композиция отношения *R* на себя), , дает нам все пути длины *2*. Напомним, что для получения *булевого произведения* , в обычной операции произведения матриц сложение заменяется *логическим* *«или»* (*дизъюнкцией*), а умножение – *логическим* *«и»* (*конъюнкцией)*. Построим :

Первая единица в первой строке соответствует ориентированному пути длины *2* из вершины *a* в вершину *c*, путь *abc*.

В общем случае, матрица хранит сведения о путях длины k. И, наконец, в матрице достижимости орграфа G = (V, E) с n вершинами, содержатся пути любой длины между вершинами .

*Матрица достижимости* определяется как:

*. (1)*

Между матрицами берется операция *логического «или»*, , определяемая для каждого элемента *матрицы достижимости* следующим образом:

*.*

Здесь обозначен элемент матрицы .

Построив последовательно матрицы , , и воспользововшись для нашего графа с *4* вершинами формулой *(1):*

,  
мы получим *матрицу достижимости*:

*Матрица достижимости* *орграфа* G = (V, E) *фактически является матрицей замыкания по транзитивности*  *отношения R* *на вершинах орграфа* G.

Напомним свойство транзитивности для отношений.

Отношениие является *транзитивным*, если для всех *а,* *b* и *c* из *А* из того, что , следует, что . Построение замыкания по транзитивности и заключается в прибавлении к *R* такихпар вида *(a,c)*.

*Для орграфа это означает, что если имется дуга, выходящая из a и входящая в b и дуга, выходящая из b и входящая в c , то существует ориентированный путь abc и вершина c достижима из a (соответствующий элемент в матрице достижимости ).*

Построить транзитивное замыкание отношения *R* можно, используя алгоритм *Уоршелла,* реализация указана ниже на *java.*

package vdm.closure;  
  
public class Main {  
  
 public static void main(String[] args) {  
 int n = 4;  
  
 boolean[][] matrix = {  
 //a, b, c, d  
 {false, true, false, false},//a  
 {false, false, true, true },//b  
 {true, false, false, true },//c  
 {false, false, false, false} //d  
  
// result:  
// true true true true  
// true true true true  
// true true true true  
// false false false false  
  
 };  
  
 for(int k = 0; k < n ; k++) {  
 for(int i = 0; i < n; i++)  
 {  
 for(int j = 0; j < n; j++)  
 {  
 matrix[i][j] = matrix[i][j] || (matrix[i][k] && matrix[k][j]);  
 }  
 }  
 }  
 System.*out*.println("result:");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println(" ");  
 }  
 }  
}

Программа написана для нашего графа с *рис.1*.

На вход алгоритма подается *матрица смежности* . На выходе алгоритма получается матрица *транзитивного замыкания* отношения *R* или, что то же самое, матрица достижимости для нашего орграфа.

Нумерация строк и столбцов начинается с 0, договоримся элементы в них называть с первого.

При каждом проходе цикла по k, начиная с k = 0, из элементов предыдущей матрицы вычисляется новая матрица. Элементы новой матрицы вычисляются по формуле:

matrix[i][j] = matrix[i][j] || (matrix[i][k] && matrix[k][j]); *(2)*

Положим k = 0 и посмотрим как строится новая матрица, обозначим ее как . Из формулы *(2)* следует,что если значение некоторого элемента в прелыдущей матрице, то его значение в новой матрице также равно *1* (это нам обеспечивает логическая операция *«или»*, ||).

Итак, k = 0, *i = 0*, строим *0* строку матрицы , индекс матрицы будем указывать в элементах сверху.

Рассмотрим операцию  *(«и»)* в скобках, мы проводим операцию между *первым* элементом нулевого столбца и всеми элементами нулевой строки *поочередно*. Эта операция строит новые пары по *замыканию*, в данном случае рассматривали бы пару *(a,a)* со всеми парами, начинающимися с *a.* Но пары *(a,a)* в нашей *текущей* матрице нет, поэтому все выражения в скобках равны *0* и нулевая строка повторяет нулевую строку из предыдущей матрицы.

Дуги *(a,a)* в нашем орграфе нет и, соответственно, нет пути, содержащего эту дугу, если говорить о *матрице достижимости*.

*Говоря в терминах графов, при каждом* k мы ищем все пути (произвольной длины), проходящие через вершину k и имеющие внутренние вершины из множества .

Строим *1* строку матрицы . Берем *второй* элемент *нулевого* столбца, , поэтому значения выражений в скобках с элементами *нулевой строки* также равны *0*, следовательно, повторяем первую строку из предыдущей матрицы. Можно сказать иначе, пары *(b,a)* в *текущей* матрице у нас тоже нет и мы не можем построить новую пару по замыканию с парой *(a, b)* из нулевой строки.

Строим *2* строку матрицы . Берем *третий* элемент *нулевого* столбца матрицы , , пара *(c,a)* у нас есть и вместе с парой *(a,b)* она нам даст пару *(c,b)*, . Больше *новых* пар по замыканию для этой строки мы построить не можем.

*И, как мы уже говорили, единицы из старой матрицы переносятся в новую.*

Если говорить об алгоритме как о построении *матрицы достижимости*, то мы построили путь *cab* и отметили, что вершина *b* доcтижима из *c.*

Строим *третью* строку матрицы . Берем *четвертый* элемент *нулевого* столбца матрицы , . Исходя из предыдущих рассуждений, третья строка переносится из матрицы .

На первом шаге алгоритма, при k = 0, построили матрицу :

Теперь обобщим наше построение.

За каждый проход цикла (пронумерованный индексом k, начиная с k=0) алгоритм *Уоршелла* генерирует матрицу , используя элементы предыдущей матрицы .

Для того, чтобы найти *i*-ую строку матрицы , нам следует вычислить  
выражения:  
 *(3)*

при разных значениях j.

Для орграфа это означает, что при прохождении цикла с номером k , мы рассматриваем k-ую вершину и все дуги, входяшие в нее и выходящие из нее и указываем наличие соответствующего ориентированного пути, проходящего через вершину k.

Теперь проанализируем выражение *(3).*

Если ( нет дуг, входящих в k из *i* ), то и значение выражения *(3)* совпадает со значением . Другими словами, *i-ая строка матрицы остается неизменной.*

Если , вычисление выражения *(3)* сводится к вычислению  
значения выражения . Это означает, что если в вершину k входит дуга из *i,* рассматриваем имеются ли дуги из k во все *j* и если для некоторого такая дуга есть, , то и соответствующий элемент матрицы =1.

*Правило обработки*.

*При этом* *i-ая строка получается с помощью логической операции*  ***(****или****)*** *из текущей строки i и текущей строки* k*.*

*Алгоритм может быть записан так:*

1. *Берем* k *–ый столбец матрицы –*
2. *Строку с номером i (), у которой на* k*-ом месте  
   стоит 0, переписываем в i -ую строку матрицы .*
3. *Для строки с номером i () , у которой на* k*-ом месте  
   стоит 1, проводим поэлементную (между элементами одного столбца) операцию* *(или)**с* k*-ой строкой, а результат записываем в i-ую строк матрицы .*

Рассмотрим теперь алгоритм на примере нашего графа, *рис.1*.

Исходная матрица – матрица смежности , начинаем ее обрабатывать.

Начинаем цикл с  *k=0*, рассатриваем *0-ой* столбец матрицы , соответствующий вершине *a* , этому столбцу соответствуют все дуги, *входяшие* в вершину *a* или по-другому, все пары элементов, в которых вторым элеметом является *a.* И рассматриваем *0* строку, ей соответствуют все дуги *выходящие* из *a* или,по-другому, пары эдементов, в которых первый элемент *a.*

Находим в *0* столбце единицу, ей соответствует дуга (пара) *(c,a)* ищем теперь *1* в *0* строке, ей соответствует дуга (пара) *(a,b).* Эти две пары дают нам путь *cab*, вершина *b* достижима из вершины *c* и в матрице элемент . Поскольку все остальные элементы столбца равны *0*, строки *0, 1, 3* записываем без изменений в матрицу .

Как видим, *2* строка матрицы получается поэлементой операцией из *0* и *2* строк матрицы , как и записано в *правиле обработки*.

Рассматриваем *первый* столбец, *k=1,* соответствующий вершине *b* ианалогично ищем пути, проходящие через вершину *b.* Воспользуемся алгоритмом и построим матрицу из .

На этом шаге добавились пути, проходящие через вершины *из* множества *{a, b}.*

Обрабатываем *второй* столбец, вершина *c.*

*k=2*

Обрабатываем *третий* столбец, вершина *d.* Поскольку из вершины *d* не выходит ни одна дуга, матрица остается неизменной и

*k=3*

Матрица является *матрицей достижимости* нашего орграфа и *матрицей транзитивного замыкания* отношения *R.*

По матрице мы бы построили *новый граф*, являющийся *транзитивным замыканием* нашего графа.

***Вопросы по обработке алгоритма.***

1. Какова *временная сложность* алгоритма?
2. Что означает в матрице равенство: ?
3. Что такое *матрица достижимости*?
4. Важен ли *порядок* обработки вершин?
5. Что означает, если при некотором *k* матрица не меняется?
6. В каких еще алгоритмах используется идея алгоритма *Уоршелла*?