Задание для курсовой работы по графам

<u>Цель курсовой работы:</u> освоение математического аппарата задания, анализа графовых моделей дискретных систем и решения ряда основных задач на графах.

Индивидуальный вариант графов необходимо получить на сайте ws-dss.com (метод **graph_generator**).

<u>Порядок выполнения:</u>

1. Описать исходные графы аналитически.
2. Построить аналитически и графически объединение, пересечения, разность графов G (X, F) и H (Y, P), дополнение графа G по отображению до графа H и до универсального графа L.
3. Для графа $S=G \cup H$ построить матрицы смежности, инцидентности, достижимости, конденсацию графа. Определить все базовые множества графа.
4. Для графа $S = G \cup H$ построить Гамильтонов (если он не существует, то дополнить граф необходимыми дугами, обозначив их на графе). Гамильтонов путь построить - с использованием алгоритма Фаулкса.
5. Получить из графа S неориентированный граф путем замены всех ориентированных ребер, на неориентированные. Лишние (параллельные) ребра удалить. Найти в полученном графе Эйлеров путь. Если он не существует, то дополнить граф необходимыми звеньями, обозначив их на графе.

Полученные результаты сравнить с результатами расчетов на ЭВМ (методы **hamiltonian_path** – для ориентированного графа и **eulerian_path** – *только для неориентированного графа*).

6. Определение связности графа.

6.1. Для заданного с помощью матрицы смежности неориентированного графа найти связные компоненты, используя алгоритм Фаулкса.

6.2. Представить заданный граф графически.

6.3. Найти связные компоненты в графе, используя метод **graph_connectivity** на сайте ws-dss.com. Сравнить полученные результаты.

7. Для рассмотренных задач:

- поиск Гамильтонова пути;
- поиск Эйлерова пути;
- определение связности графа;
- поиск базовых множеств,

предложить содержательное (предметное) описание подходящей технической задачи.

8. Оформить отчет. Отчет должен содержать содержание, задание, ход выполнения работы с пояснениями, выводы по работе, список используемой литературы.

Методические указания

При проектировании различных систем часто приходится решать задачи, которые можно назвать задачами согласования и упорядочения. Такие задачи, в частности, возникают при планировании работы цехов и предприятий, выпускающих широкую номенклатуру продукции с использованием в производственном процессе различных комбинаций станков и другого оборудования. Даже если используется всего одна единица оборудования, на которой выполняются различные работы, задача определения порядка выполнения работ является часто довольно сложной. Аналогичные проблемы возникают при определении очередности решения задач на ЭВМ; при определении наличия взаимосвязей между группами объектов в сложной информационной системе; при определении подчиненности одних элементов системы другим, возможности передачи управляющих воздействий (команд) ко всем элементам сложной системы и т.д.

Решение многих из них эффективно сводятся к решению известных задач с использованием аппарата теории графов, в частности, таких как поиск гамильтонова пути в графе, определение связности графа, поиск эйлерового пути в графе, определение основных элементов сложной структуры - сильных компонент, базового и доминирующих множеств в графе и других.

Основные понятия. Способы задания графа

Под **графом** будем понимать систему некоторых абстрактных объектов вместе с некоторыми парами из этих объектов, для которых заданы отношения связи между ними.

Граф **G** задается множеством точек (вершин) $\mathbf{X}\{x_1,x_2,\ldots,x_n\}$ в **n**-мерном пространстве и множеством кривых (ребер) $\mathbf{U}\{u_1,u_2,\ldots,u_n\}$, соединяющих между собой все или часть точек, то есть граф задается в виде $\mathbf{G(X,U)}$, где **X** – множество вершин графа, **U** – множество ребер графа.

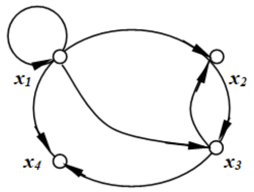
Если ребра графа ориентированные, то такой граф называется ориентированным, а его ребра называются дугами. Замкнутая дуга называется петлёй.

Если ребра графа неориентированные, то такой граф называется неориентированным, а его ребра называются звеньями.

Аналитическое задание графа

Ориентированный граф G = (X, F) есть совокупность двух объектов: множества вершин $X=\{x_1,x_2,\ldots,x_n\}$и отображения $F=\{Fx_i \subseteq X\}$, характеризующего связи между вершинами. Отображение Fx_i указывает – к каким элементам множества X ведут дуги, исходящие из элемента x_i (i = 1,..,n).

Например: Граф G = (X, F).



Здесь:

$X=\{x_1, x_2, x_3, x_4\}$;

$Fx_1=\{x_1, x_2, x_3, x_4\}$; $Fx_2=\{x_3\}$; $Fx_3=\{x_2, x_4\}$; $Fx_4=\varnothing$.

Матричные способы задания графов

I. Матрица смежности R ориентированного графа G=(X,F) есть квадратная (n×n) матрица **R={rij}** i,j=1..n, строки и столбцы которой соответствуют вершинам графа, а элементы матрицы r_{ij} – булевые переменные и равны соответственно: 1 если $x_j \in Fx_i$, то есть существует дуга (x_i, x_j), 0 если $x_j \notin Fx_i$, т.е. дуга (x_i, x_j) – не существует.

Для рассмотренного выше графа G = (X, F) матрица смежности R будет

	x_1	x_2	x_3	x_4
x_1	1	1	1	1
x_2	0	0	1	0
x_3	0	1	0	1
x_4	0	0	0	0

Для неориентированного графа элементы матрицы смежности $r_{i,j}$ будут равны: 1, если есть ребро (звено) соединяющее x_i с x_j ; 0, если такое ребро отсутствует.

II. Матрица инцидентности A ориентированного графа G=(X, F) есть прямоугольная матрица **A={a_{ik}}**, строки которой соответствуют вершинам графа, а столбцы - его ребрам. Тогда элементы матрицы a_{ik} будут равны:

0 – если вершина x_i не является граничной точкой для ребра u_k ;

+1 – если дуга u_k заходит в вершину x_i ;

−1 – если дуга u_k исходит из вершины x_i

Для рассмотренного выше графа G = (X, F) матрица инцидентности **A** будет

	u_1	u_2	u_3	u_4	u_5	u_6	u_7
x_1	-1	0	0	-1	0	-1	1
x_2	+1	-1	+1	0	0	0	0
x_3	0	+1	-1	+1	-1	0	0
x_4	0	0	0	0	+1	+1	0

Операции над графами

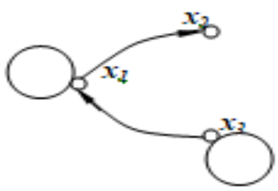
На основе теоретико-множественной интерпретации графов становится возможным определить на них операции, аналогичные, в некотором смысле, ряду операций над множествами.

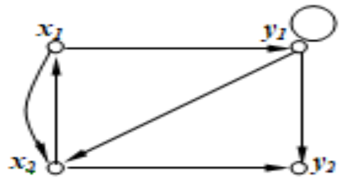
<u>Объединение графов</u>

Граф Q(A,S) называется **<u>объединением</u>** графов G(X,F) и H(Y,P), если $A=X\cup Y$, и для любых $a\in A$, $S_a = F_a \cup P_a$, причем для $a\notin X$ $F_a=\varnothing$, и для $a\notin Y$ $P_a=\varnothing$. Граф Q получается объединением всех вершин и ребер графов G и H. То есть **Q = G∪H**.

Операция объединения может быть обобщена для нескольких графов.

<u>Пример</u>:



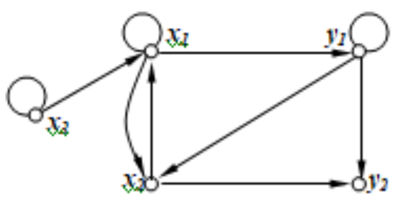


<u>G(X,F)</u> :
$X=\{x_1, x_2, x_3\}$
$Fx_1=\{x_1, x_2\}$
$Fx_2=\varnothing$
$Fx_3=\{x_1, x_3\}$

<u>H(Y,P)</u> :
$Y=\{x_1, x_2, y_1, y_2\}$
$Px_1=\{x_2, y_1\}$
$Px_2=\{x_1, y_2\}$
$Py_1=\{x_2, y_1, y_2\}$
$Py_2=\varnothing$

<u>Q =G∪H = Q(A,S):</u>

$A=X\cup Y = \{x_1, x_2, x_3\} \cup \{x_1, x_2, y_1, y_2\} = \{x_1, x_2, x_3, y_1, y_2\}$;
$Sx_1= F_{X1}\cup P_{X1} = =\{x_1, x_2\} \cup \{x_2, y_1\} = \{x_1, x_2, y_1\}$;
$Sx_2= F_{X2}\cup P_{X2} = \varnothing \cup \{x_1, y_2\} = \{x_1, y_2\}$;
$Sx_3= F_{X3}\cup P_{X3} = \{x_1, x_3\} \cup \varnothing = \{x_1, x_3\}$;
$Sy_1= F_{Y1}\cup P_{Y1} = \varnothing\cup\{x_2, y_1, y_2\} = \{x_2, y_1, y_2\}$;
$Sy_2= F_{Y2}\cup P_{Y2} = \varnothing \cup \varnothing = \varnothing$.



<u>Пересечение графов</u>

Граф Q(A,S) называется **<u>пересечением</u>** графов G(X,F) и H(Y,P), если $A=X\cap Y$ и для любых $a\in A$, $S_a=F_a \cap P_a$, причем для $a\notin X$ $F_a=\varnothing$, и для $a\notin Y$ $P_a=\varnothing$. Граф Q получается в результате выделения только тех вершин и ребер, которые одновременно содержатся и в графе G и в графе H. То есть **Q = G∩H**.

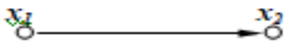
Операция пересечения графов также может быть обобщена и для нескольких графов.

Для предыдущего примера. Даны графы $G=(X, F)$, $H=(Y, P)$:

Граф $Q = G \cap H = Q(A, S)$:

$A = X \cap Y = \{x_1, x_2, x_3\} \cap \{x_1, x_2, y_1, y_2\} = \{x_1, x_2\}$;
$Sx_1 = F_{X1} \cap P_{X1} = = \{x_1, x_2\} \cap \{x_2, y_1\} = \{x_2\}$;
$Sx_2 = F_{X2} \cap P_{X2} = \varnothing \cap \{x_1, y_2\} = \varnothing$;

Граф $Q = G \cap H$ имеет вид:



Если в результате пересечения двух графов образуется пустой граф Λ (порядка n=0 с пустым отображением), то графы называются **непересекающимися**.

Разность графов

Граф Q(A,S) называется **разностью** графов G(X,F) и H(Y,P), если A=X, и для любых $a \in A$ $(a \in X)$, $S_a = F_a \setminus P_a$. То есть $Q(A,S) = G(X,F) \setminus H(Y,P) = Q(X, (F_x \setminus P_x))$.

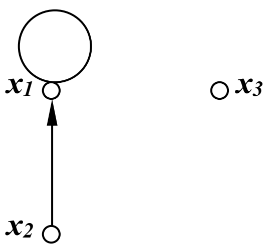
Пример:

G(X,F) :
$X=\{x_1, x_2, x_3\}$
$Fx_1=\{x_1, x_3\}$
$Fx_2=\{x_1\}$
$Fx_3= \varnothing$

H(Y,P) :
$Y=\{x_1, x_3, x_4\}$
$Px_1=\{x_3\}$
$Px_3=\{x_3, x_4\}$
$P x_4=\varnothing$

$Q(A,S) = .G(X, F) \setminus H(Y, P) = Q(X, (F_x \setminus P_x))$:

$A = X = \{x_1, x_2, x_3\}$;
$Sx_1 = F_{X1} \setminus P_{X1} = = \{x_1, x_3\} \setminus \{x_3\} = \{x_1\}$;
$Sx_2 = F_{X2} \setminus P_{X2} = \{x_1\} \setminus \varnothing = \{x_1\}$;
$Sx_3 = F_{X3} \setminus P_{X3} = \varnothing \setminus \{x_3, x_4\} = \varnothing$.

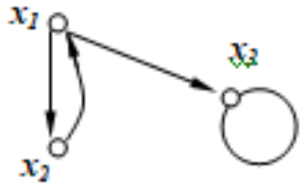


Дополнение графа по отображению

Операция **дополнения графа G по отображению до графа H** может быть определена для графов **только тогда**, когда рассматриваются два графа G и H такие, что граф G является подграфом графа H ($G \subseteq H$).

В этом случае дополнением по отображению графа G до графа H ($G=(X,F)$, $H=(Y,P)$) называется такой граф $\overline{G}_H(Y,\overline{F})$, для которого для любых $y \in Y$, отображение $\overline{F}_y = P_y \backslash F_y$. Причем, если $y \notin X$, то $F_y = \varnothing$, отсюда $\overline{F}_y = P_y \backslash \varnothing = P_y$. Таким образом граф $\overline{G}_H$ задается на всем множестве вершин графа H (то есть на Y) и включает только те ребра, которые есть в графе H, но отсутствуют в графе G.

Пример: Даны графы G и H, ($G \subseteq H$).



G(X,F):
$X=\{x_1, x_2\}$
$Fx_1=\{x_2\}$
$Fx_2=\varnothing$

H(Y,P):
$Y=\{x_1, x_2, x_3\}$
$Px_1=\{x_2, x_3\}$
$Px_2=\{x_1\}$
$Px_3=\{x_3\}$

Граф $\overline{G}_H = \overline{G}_H(Y,\overline{F})$:
$Y=\{x_1, x_2, x_3\}$

$\overline{F}_{X1} = Px_1 \backslash Fx_1 = \{x_2,x_3\} \backslash \{x_2\} = \{x_3\}$;
$\overline{F}_{X2} = Px_2 \backslash Fx_2 = \{x_1\} \backslash \{\varnothing\} = \{x_1\}$;
$\overline{F}_{X3} = Px_3 \backslash Fx_3 = \{x_3\} \backslash \{\varnothing\} = \{x_3\}$.

Граф $\overline{G}_H$ имеет вид:

Наряду с дополнением графа G по отображению до некоторого произвольного графа H ($G \subseteq H$) на практике используют **дополнение графа G по отображению до универсального графа L**.

Под **универсальным графом L** понимается такой граф, для которого все графы, рассматриваемые в данном круге задач, являются подграфами. Таким графом является **n**-мерный насыщенный граф, т.е. граф, имеющий **n** вершин и все ребра, включая петли. Множество вершин такого графа можно получить, в частности, путем объединения множеств вершин рассматриваемых графов.

Дополнение графа G по отображению до универсального графа L(Z,Z) обозначается как $\overline{G} = \overline{G}(Z,\overline{F})$. Оно задается на всем множестве вершин **Z** универсального графа, и имеет отображение $\overline{F}$.

Примечание: С помощью операций пересечения и дополнения по отображению до универсального графа может быть получена разность графов. В этом случае **разностью графов** G(X,F) и H(Y,P) будет граф:

Q(A,S) =.G(X, F) \ H(Y, P) = G(X, F) $\cap$ $\overline{H}(Z,\overline{P})$,

где $\overline{H}(Z,\overline{P})$ – дополнение графа H по отображению до универсального графа L.

Основные задачи на графах

Определение достижимости вершин графа

Будем полагать, что вершина $\mathbf{x_j}$ **достижима** из вершины $\mathbf{x_i}$, если существует хотя бы один путь, ведущий из вершины x_i к вершине x_j.

Достижимость вершин в графе может быть задана с помощью **матрицы достижимости** **L={l_{ij}}**, элементы которой l_{ij} равны: 1 если вершина $\mathbf{x_j}$ достижима из $\mathbf{x_i}$, **0**, если вершина $\mathbf{x_j}$ недостижима из $\mathbf{x_i}$. Очевидно, что все элементы, стоящие на главной диагонали в матрице достижимости равны 1 (каждый элемент достижим сам для себя).

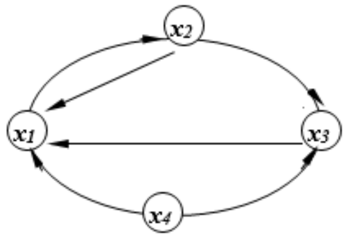
Нахождение матрицы достижимости:

Матрица достижимости L может быть достаточно просто получена из матрицы смежности R графа с использованием операций логического сложения и умножения.

Для этого берется матрица R^1 (матрица достижимости не более чем за один шаг), которая получается из матрицы смежности R исходного графа путём расстановки единиц на главной диагонали (т.к. каждая вершина достижима сама для себя).

Используя операции логического булевого $\otimes$ умножения матриц получим матрицу $R^2 = R^1 \otimes R^1$, которая соответствует наличию путей длиной не более 2-х от вершины $\mathbf{x_i}$ к вершине $\mathbf{x_j}$. Затем $R^3 = R^2 \otimes R^1$ – пути длиной ≤ 3 и т.д., до тех пор, пока последующая матрица не перестанет отличаться от предыдущей. Полученная матрица есть матрица достижимости **L** исходного графа G.

Пример: Дан граф G=(X,F).



Ему соответствуют матрица смежности **R** и матрица достижимости не более чем за один шаг $\mathbf{R^1}$:

$$\mathbf{R} = \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{matrix}, \quad \mathbf{R^1} = \begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix}.$$

Найдем матрицу достижимости:

$$\mathbf{R^2 = R^1 \otimes R^1} = \begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix} \otimes \begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix} = \begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix},$$

$$\mathbf{R^3 = R^2 \otimes R^1} = \begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix} \otimes \begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix} = \begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix} = \mathbf{R^2}$$

Так как $\mathbf{R^2 = R^3}$, то полученная матрица $\mathbf{R^3}$ есть матрица достижимости **L**.

Нахождение матрицы контрдостижимости:

В отличие от матрицы достижимости **L**, **матрица контрдостижимости** $\mathbf{Q}=\{q_{ij}\}$ $i,j=1..n$ характеризует, из каких вершин x_j может быть достигнута вершина x_i. То есть элемент матрицы $\mathbf{q_{ij}}$ равен: 1 если вершина $\mathbf{x_i}$ достижима из $\mathbf{x_j}$; **0**, если вершина $\mathbf{x_i}$.недостижима из $\mathbf{x_j}$ **.** Матрица контрдостижимости Q ориентированного графа может быть получена путём транспонирования матрицы достижимости, то есть: $\mathbf{Q = L^T}$ **.**

Для рассмотренного выше примера матрица контрдостижимости **Q** равна:

$$\mathbf{Q = L^T} = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Так как для неориентированного графа матрица смежности R равна R^T , поэтому для него матрицы достижимости **L** и контрдостижимости **Q** будут совпадать.

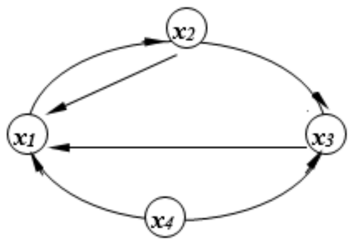
Определение сильных компонент графа

Ориентированный граф H(Y,P) **сильно связный**, если для любой пары вершин (y_i, y_j) существует хотя бы один путь как из $\mathbf{y_i}$ в $\mathbf{y_j}$,так и наоборот из $\mathbf{y_j}$ в $\mathbf{y_i}$. То есть любые две вершины взаимно достижимы.

Под **<u>сильной компонентой (СК)</u>** графа **G**(X,F) будем понимать подграф **G'=СК** графа **G** максимальной размерности (максимальное число вершин), являющийся сильно связанным графом, который не содержится в любом другом сильно связном подграфе графа **G**.

Для нахождения сильных компонент графа **G** произведём поэлементное логическое перемножение (●) матриц достижимости **L** и контрдостижимости **Q** . В результате получим матрицу СК=L●Q. Наличие 1 в i-й строке матрицы СК говорит о взаимной достижимости вершины $\mathbf{x_i}$ с вершинами x_j, для которых элемент строки $\mathbf{x_i}$ равен единице. Следовательно, эти вершины, включая $\mathbf{x_i}$ **,** входят в сильную компоненту $\mathbf{СКx_i}$. Меняя местами строки и столбцы матрицы можно привести ее к блочно-диагональному виду, где каждая из диагональных подматриц, состоящих из единиц, соответствует одной из сильных компонент графа G. Все остальные элементы матрицы равны нулю и соответствующие им вершины не являются взаимодостижимыми.

<u>*Пример*</u>: граф G=(X,F).



Ему соответствуют матрица достижимости L и контрдостижимости Q.

$$L = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad Q = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Перемножив их поэлементно получим матрицу $\mathbf{СК = L \bullet Q =} \begin{matrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix}$.

Таким образом граф G разбит на свои сильные компоненты $СКx_1$, и $СКx_2$: $СК_1=\{x_1,x_2,x_3\}$, $СК_2=\{x_4\}$.

На этих сильных компонентах , как на вершинах, может быть построен новый граф G*(X*,F*), который называется **<u>конденсацией</u>** <u>графа G</u>. Каждая его вершина соответствует некоторой сильной компоненте графа G, а дуга (x_ξ^*,x_η^*) существует в конденсации G* только тогда, когда в графе G существует дуга (x_i,x_j) такая, что $x_i \in СКx_\xi$, а $x_j \in СКx_\eta$.

В нашем случае конденсация графа включает в себя две вершины $x_1^*=\{x_1,x_2,x_3\}$ и $x_2^*=\{x_4\}$. С учетом связей в исходном графе между вершинами сильных компонент, конденсация будет иметь вид:

Определение базы графа

<u>База B</u> графа G=(X,F) есть множество **B** вершин графа G, из которых достижима любая вершина графа и которое является минимальным в том смысле, что не существует такого подмножества множества **B**, обладающего таким же свойством достижимости.

<u>*Для нахождения базы графа G*</u> можно сделать следующее:

- построить конденсацию G*=(X*,F*) графа G, каждая из вершин которой соответствует сильной компоненте графа G;
- построить базу B* для конденсации G*, которая включает те вершины x_ξ^* конденсации G*, полустепени захода которых равны нулю;
- из каждой сильной компоненты, соответствующей вершине $x_\xi^* \in B^*$ необходимо взять по одной (любой) вершине $\mathbf{x_{i\xi} \in x_\xi^*}$ (ξ-й СК графа G).

Множество таких вершин $\mathbf{x_{i\xi}}$ и образует базу графа G(X,F).

В нашем примере базой B* конденсации СК графа G будет вершина x_2^* и, следовательно, <u>базой исходного графа G=(X,F)</u> будет B={x_4}.

Задача нахождения базового множества в графе эквивалентна задаче нахождения минимальной ДНФ на импликантной матрице, т.е. такого наименьшего множества строк, что каждый столбец матрицы содержит единицу, хотя бы в одной из этого множества строк. Только вместо импликантной матрицы, при поиске базового множества используется матрица достижимости L графа G.

Таким образом, задача нахождения базового множества в графе сводится к определению такого множества строк матрицы достижимости (вершин графа), из которых достижима любая вершина графа и, при этом, из этого множества нельзя исключить ни одной вершины, без нарушения свойства достижимости всех вершин графа.

<u>Путь в графе</u>

Путем (или **ориентированным маршрутом**) в ориентированном графе называется последовательность дуг, в которой конечная вершина всякой предыдущей дуги (отличной от последней) является начальной вершиной последующей дуги. **Маршрут в ориентированном графе** – такой путь, в котором направление дуг не учитывается.

Путь имеет начальную и конечную вершины. **Длина пути** – число дуг (рёбер), составляющих путь.

Простой путь – путь, в котором никакая дуга не встречается дважды. **Элементарный путь** – путь, в котором никакая вершина не встречается дважды.

Нахождение гамильтонова пути в графе

Гамильтонов путь в графе – это путь, проходящий через все вершины графа, причем каждая вершина встречается в нем только один раз. Если этот путь замкнут, то говорят о гамильтоновом цикле.

В качестве примера рассмотрим задачу определения очередности решения на ЭВМ вычислительной программы (алгоритма) **А**, состоящей из ***n*** подпрограмм A_i ; $i=1,2,\ldots,n$, причем на порядок выполнения подпрограмм накладываются ограничения связи подпрограмм вида:

$A_i \rightarrow A_j$, т.е. выполнение подпрограммы A_i предшествует выполнению подпрограммы A_j , $i, j \in \{1,2,\ldots,n\}$, или

$A_i \leftrightarrow A_j$, т.е. порядок выполнения программ A_i и A_j безразличен.

Эту задачу удобно геометрически отобразить в виде графа, где каждой **i** вершине соответствует A_i - подпрограмма, а связи между подпрограммами соответствуют дугам или ребрам. Причем дуга соответствует связи $A_i \rightarrow A_j$, а ребро - связи $A_i \leftrightarrow A_j$.

Если предполагается, что программа А будет выполнена, когда выполнены все подпрограммы A_i , $i=1,2,\ldots,n$ (без повторений), то определение очередности выполнения подпрограмм A_i сводится к определению на графе гамильтонова пути. Очевидно, что эту задачу можно решить перебором всех возможных вариантов ($n!$ в насыщенном графе) с одновременной проверкой выполнения всех условий связи. При $n = 2$ максимально возможно всего два варианта, при $n = 3$ - шесть, при $n=10$ существует более трех с половиной миллионов возможных вариантов, из которых надо выбрать допустимые, удовлетворяющие условиям связи.

Решение поставленной задачи может быть облегчено, если воспользоваться алгоритмом Фаулкса, позволяющим во многих случаях значительно уменьшить количество рассматриваемых возможных вариантов [1].

Алгоритм Фаулкса. Рассматривается ***n*** операций (вершины графа $A_1, A_2, \ldots, A_n$), между которыми существуют соотношения связи: $A_i \rightarrow A_j$, $A_i \rightarrow A_j$, $i, j \in \{1,2,\ldots,n\}$.

Задача состоит в том, что среди всех возможных путей найти, если это возможно, путь (или несколько путей), проходящий один и только один раз через каждую из вершин графа и удовлетворяющий заданным выше соотношениям, то есть гамильтонов путь.

Идея вычислений, которые необходимо выполнить при использовании алгоритма Фаулкса, заключается в нахождении матрицы R^N путем последовательного булевого умножения $\otimes$ матрицы R^1 (достижимости не более чем за один шаг) саму на себя N раз.

$$R^N = \underbrace{R^1 \otimes R^1 \otimes ... \otimes R^1}_{N}$$

Элементы r^N_{ij} матрицы R^N**,** определяются как

$$r_{ij}^N = \bigvee_{k=1}^{n} r_{ik}^{N-1} \cdot r_{kj}$$

Вычисления следует прекратить, если $R^N = R^{N-1}$.

На каждом цикле вычисления возможны некоторые упрощения матрицы, связанные с выявлением «*начальных*» и «*конечных*» вершин, ограничивающих возможность дальнейшего построения ГП из общего множества возможных путей в графе.

Порядок решения задачи:

1. Задаем номер цикла вычислений N=1.

2. Используя заданные соотношения связи, составляем граф, вершины которого есть операции, а направленные дуги графа определяются заданными соотношениями между операциями.

3. Составляем матрицу R^1 графа, элементы r^1_{ij} которой могут принимать значения 0 либо 1. Если $r^1_{ij} = 1$, то это указывает на то, что операция (вершина) A_j должна следовать за операцией (вершиной) A_i . Полученная таким образом матрица R^1 является булевой матрицей (каждый элемент может принимать значения 0 или 1). Она является исходной для всех дальнейших вычислений.

4. Используя матрицу R^N (на первом цикле вычислений $R^N = R^1$)*,* для каждой вершины графа (операции) проверяем принадлежность ее к начальной или конечной вершине ГП по следующим правилам.

Вершина (операция) является *начальной* вершиной ГП, если во всей строке матрицы стоят единицы, а во всем столбце (за исключением их пересечения) - нули. Вершина (операция) является *конечной* вершиной ГП, то во всей строке матрицы стоят нули (за исключением их пересечения), а во всем столбце - единицы.

5. Упрощаем матрицу R^N графа, вычеркивая из нее строки и столбцы, соответствующие начальной или (и) конечной вершине ГП. Получаем матрицу $(R^N)'$.

6. Увеличиваем на единицу номер цикла, т.е. N=N+1.

7. Находим матрицу $(R^N)'$, выполнив булевое умножение матрицы $(R^{N-1})'$ на матрицу $(R^1)'$, которая получается из исходной матрицы R^1 вычеркиванием из нее строк и столбцов, соответствующих начальным и конечным вершинам ГП, найденным во всех предыдущих циклах вычислений. Булевое перемножение матриц производится по обычным правилам умножения, только вместо умножения используется конъюнкция, а вместо сложения – дизъюнкция.

8. Проверяем условие выполнения равенства матриц $(R^N)'$ и $(R^{N-1})'$.

Если это условие выполняется, то переходим к п. 10, Иначе - к п. 9.

9. Проверяем условие равенства всех элементов матрицы R^N единице. Если это условие выполняется, то нет необходимости в дальнейших вычислениях матрицы $(R^N)'$, так как очевидно, что $(R^{N+1})' = (R^N)'$, переходим к п. 10. Если нет - к п. 4.

10. Составляем матрицу R^b, возвращая в матрицу $(R^N)'$, строки и столбцы, вычеркнутые в п. 5 во всех циклах вычислений.

11. Составляем матрицу R^c, перегруппировывая одновременно строки и столбцы матрицы R^b таким образом, чтобы все нули были расположены под главной диагональю, а единицы - над ней.

Квадратные матрицы, состоящие из единиц, опирающихся на главную диагональ, образуют классы эквивалентности вершин (совокупности вершин графа, эквивалентных с точки зрения очередности их присутствия в гамильтоновом пути).

Если для данного графа мы получили m классов эквивалентности $B_1,\ldots,B_m$ $(m \le n)$, и в каждый класс эквивалентности B_d, $d = 1,2,\ldots,m$ входят S_d вершин графа, то можно сказать, что вершины, входящие в класс эквивалентности B_d, расположенные выше и левее класса эквивалентности B_l, в матрице R^c будут предшествовать в Гамильтоновом пути вершинам из класса эквивалентности B_l.

Нахождение Гамильтоновых путей в этом случае значительно упрощается. В соответствии с классами эквивалентности мы получили упорядоченные группы вершин. В качестве начальной выбирается вершина из первого класса эквивалентности. Если их в нем несколько, то следующая выбирается из этого же класса и так до тех пор, пока не будут исчерпаны все вершины этого класса. Затем переходим к вершинам следующего класса эквивалентности и так далее, пока не будут использованы все вершины графа и построен ГП. Следует заметить, что при выборе очередной вершины необходимо обязательно проверять выполнение условий связи. Если условия не выполняются, очередность следования вершин из одного класса эквивалентности меняется. Максимальное число возможных вариантов сокращается до произведения $B_1! \cdot B_2! \cdot ,\ldots, \cdot B_m!$

Пример. Пусть программа A состоит из 6 операций (подпрограмм) $A_1, A_2, \ldots, A_6$, между которыми существуют следующие соотношения связи:

$A_1 \to A_2$, $A_2 \to A_3$, $A_3 \to A_4$, $A_5 \to A_2$, $A_6 \to A_2$,

$A_1 \to A_4$, $A_2 \to A_4$, $A_5 \to A_4$, $A_6 \to A_4$,

$A_1 \to A_6$, $A_2 \to A_5$, $A_6 \to A_5$

$A_2 \to A_6$.

Цель задачи состоит в нахождении пути, проходящего только один раз через все операции (вершины графа) и удовлетворяющего написанным соотношениям. Без учета ограничений связи всего возможно 6! = 720 вариантов следования вершин.

<u>Для решения задачи воспользуемся приведенным алгоритмом.</u>

1. Зададим номер цикла $N = 1$.

2. Составим ориентированный граф (см. рис.), соответствующий заданным выше соотношениям связи.

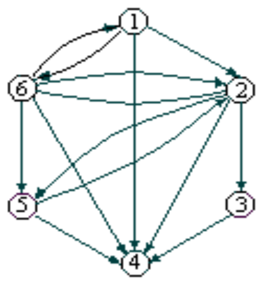


Рис. Исходный граф для поиска ГП.

3. Составим матрицу R^1 достижимости не более чем за один шаг.

$R^1 =$

	A_1	A_2	A_3	A_4	A_5	A_6
A_1	1	1	0	1	0	1
A_2	0	1	1	1	1	1
A_3	0	0	1	1	0	0
A_4	0	0	0	1	0	0
A_5	0	1	0	1	1	0
A_6	1	1	0	1	1	1

4. Проверим матрицу R^1 на наличие начальной или конечной вершин в графе.

Очевидно, что A_4 есть конечная вершина каждого гамильтонова пути, поскольку никакая дуга не имеет эту вершину своим началом, тогда как дуга, исходящая из любой другой вершины, достигает вершину A_4. Это свойство выражается наличием единиц во всем столбце A_4 и нулей во всей строке A_4 (за исключением их пересечения).

5. Вычеркиваем столбец и строку A_4. Получаем матрицу $(R^1)'$.

$(R^1)' =$

	A_1	A_2	A_3	A_5	A_6
A_1	1	1	0	0	1
A_2	0	1	1	1	1
A_3	0	0	1	0	0
A_5	0	1	0	1	0
A_6	1	1	0	1	1

6. N=1+1=2.

7. Произведем булевое умножение матриц $(R^1)' \otimes (R^1)' = (R^2)'$.

Наличие 1 в матрице означает существование между соответствующими вершинами путей длиной меньшей или равной двум, а 0 - их отсутствие.

$$(R^2)' = \begin{array}{c|ccccc} & A_1 & A_2 & A_3 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 & 1 \\ A_3 & 0 & 0 & 1 & 0 & 0 \\ A_5 & 0 & 1 & 1 & 1 & 1 \\ A_6 & 1 & 1 & 1 & 1 & 1 \end{array}$$

8. Проверяем условие равенства $(R^2)'$ и $(R^1)'$. Так как они не равны, переходим к п. 9.

9. Проверяем условие равенства всех элементов матрицы единице. Так как оно не выполняется, то возвращаемся к п.4.

4. Рассмотрим матрицу $(R^2)'$.

Вершина A_3, является конечной вершиной гамильтонова пути на данном цикле вычислений.

5. Вычеркиваем столбец и строку A_3. Оставшаяся матрица $(R^2)'$ = имеет вид

$$\begin{array}{c|cccc} & A_1 & A_2 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 \\ A_5 & 0 & 1 & 1 & 1 \\ A_6 & 1 & 1 & 1 & 1 \end{array}.$$

6. N=2+1=3.

7. Производим умножение $(R^2)' \otimes (R^{1}{}') = (R^3)'$.

$$(R^3)' = \begin{array}{c|cccc} & A_1 & A_2 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 \\ A_5 & 1 & 1 & 1 & 1 \\ A_6 & 1 & 1 & 1 & 1 \end{array}.$$

8. $(R^3)' \neq (R^2)'$.

9. Все элементы $(R^N)'$, равны единице. Очевидно, что матрицы $(R^4)'$, $(R^5)'$ и т.д. также будут равны, так как будут иметь все элементы, равные единице.

Переходим к п.10.

10. Матрицу $\boldsymbol{R}^{\mathbf{b}}$ получаем возвращением строк и столбцов, вычеркнутых в п. 5.

$$
R^b = \begin{array}{c|cccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\
\hline
A_1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_3 & 0 & 0 & 1 & 1 & 0 & 0 \\
A_4 & 0 & 0 & 0 & 1 & 0 & 0 \\
A_5 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

11. Перегруппируем строки и столбцы матрицы $\boldsymbol{R}^{\mathbf{b}}$ так, чтобы все нули были расположены под главной диагональю, а единицы - над ней: Получаем матрицу $\boldsymbol{R}^{\mathbf{c}}$ **.**

$$
R^c = \begin{array}{c|cccc|cc}
 & A_1 & A_2 & A_5 & A_6 & A_3 & A_4 \\
\hline
A_1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_5 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1 & 1 & 1 \\
\cline{2-5}
A_3 & 0 & 0 & 0 & 0 & \boxed{1} & 1 \\
A_4 & 0 & 0 & 0 & 0 & 0 & \boxed{1}
\end{array}
$$

Таким образом, получили три класса эквивалентности:

- в первый класс входят вершины A_1, A_2, A_5, A_6;

- во второй - вершина A_3;

- в третий - вершина A_4.

Т.е. в гамильтоновом пути вершины A_1, A_2, A_5, A_6 предшествуют вершине A_3, которая, в свою очередь, предшествует вершине A_4. Всего возможно $4! \cdot 1! \cdot 1! = 24$ вариантов следования вершин.

С учетом необходимости выполнения условий связи, определение гамильтонова пути становится достаточно простым. В данном примере получим единственный **ГП:** $\boldsymbol{A_1, A_6, A_5, A_2, A_3, A_4}$.

<u>Нахождение эйлерового пути в графе</u>

Одна из важных задач теории графов заключается в нахождении пути (маршрута), содержащего по одному разу все ребра графа. Этот путь называется **эйлеровым путем (ЭП)** или эйлеровой линией в графе, а задачу его нахождения называют задачей Эйлера. Графы, на которых существуют замкнутые эйлеровы линии, называют **эйлеровыми графами**.

На практике задача нахождения ЭП на графе может встретиться, например, при определении порядка проведения регламентных проверок исправности каналов связи между звеньями какой-нибудь достаточно сложной системы, обслуживаемой одной ремонтной единицей.

Ответ на вопрос о существовании ЭП на графах дают теоремы Эйлера [1, 6].

Теорема Эйлера для неориентированных графов: для того чтобы в графе существовал замкнутый эйлеров путь (маршрут), необходимо и достаточно, чтобы граф был связным и степени всех его вершин были четными.

Теорема Эйлера для ориентированных графов: для того чтобы на ориентированном графе существовал замкнутый ЭП, необходимо и достаточно, чтобы граф был связным и для каждой его вершины полустепень исхода равнялась полустепени захода.

Для нахождения ЭП в неориентированном графе (если он существует) удобно воспользоваться следующим алгоритмом:

0. Задается начальный шаг итерации $k=0$. Задается начальная (она же и конечная в случае замкнутого ЭП) вершина ЭП V_0. Если, например, $V_0 = A_3$, то это означает, что начальной вершиной ЭП на шаге $k = 0$ выбрана третья вершина графа. Все ребра графа считаются непомеченными. Число непомеченных ребер S равно числу ребер графа.

1. $k=k+1$. Определяются все вершины, непосредственно связанные непомеченными ребрами с вершиной V_{k-1}.

2. Определяется число S_k непомеченных ребер, инцидентных вершине V_{k-1}.

Если $S_k= 1$, то вторая вершина, инцидентная этому ребру, становится очередной вершиной V_k ЭП. Это ребро помечается. Возвращаемся на п. 1.

Если $S_k >1$, то среди вершин, смежных V_{k-1}, выбирается вершина с наименьшим номером k, которая становится очередной вершиной V_k ЭП. Ребро, соединяющее эту вершину с вершиной V_{k-1}, помечается. Возвращаемся на п. 1.

3. $S_k = 0$. Для вершины, принадлежащей полученному таким образом пути и имеющей непомеченные инцидентные ребра, строится цикл по непомеченным ребрам. Определение новой вершины этого цикла начинается с п. 1.

4. Построенный таким образом новый цикл объединяется с ранее построенным в вершине, с которой начиналось построение нового цикла.

5. Если в цикле есть еще вершины, имеющие непомеченные инцидентные ребра, то возвращаемся на п. 3. Если таких ребер нет, то ЭП найден.

Отметим, что выбор вершины с наименьшим номером нужен для того, чтобы была определенность при выборе вершины на каждом шаге. Можно, однако, предложить выбор вершин и с наибольшим номером.

В качестве примера рассмотрим задачу нахождения ЭП для графа (см. рис.). ЭП существует, так как степени всех вершин четные.

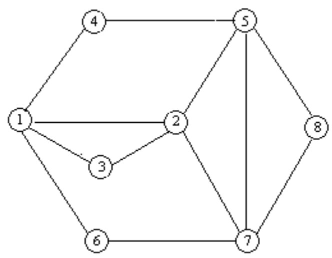


Рис. Исходный граф для поиска ЭП.

0. k = 0. Выберем начальную вершину с номером 3, т.е. V_0 = 3. Число непомеченных ребер S = 12.

1. k = 1. С вершиной V_0 = 3 связаны вершины 1, 2.

2. Число непомеченных ребер для 3-й вершины S_1 = 2. Следующей вершиной ЭП является вершина с номером 1. Ребро 3-1 помечается. Возвращаемся к п. 1.

1. k = 2. С вершиной 1 связаны вершины 2, 4, 6.

2. S_2 = 3. Вершина 2. Помечается ребро 1-2. Возвращаемся к п. 1.

1. k = 3. С вершиной 2 связаны вершины 3, 5, 7.

2. S_3 = 3. Вершина 3. Помечается ребро 2-3. Возвращаемся к п. 1.

1. k = 4. С вершиной 3 связаны вершины 1, 2.

2. S_4 = 0. Вершин таких нет. Следовательно, получен цикл 3-1-2-3, который еще не является ЭП.

3. Вершины 1 и 2, входящие в этот цикл, имеют непомеченные инцидентные ребра. Рассмотрим вершину 1 в качестве начальной точки следующего цикла A_4 = 1.

1. k = 5. Вершины 4, 6.

2. S_5 = 2. Вершина 4. Ребро 1-4.

1. k = 6. Вершина 5.

2. S_6 = 1. Вершина 5. Ребро 4-5.

1. k = 7. Вершины 2, 7, 8.

2. S_7 = 3. Вершина 2. Ребро 5-2.

1. k = 8. Вершина 7.

2. S_8 = 1. Вершина 7. Ребро 2-7.

1. k = 9. Вершины 5, 6, 8.

2. S_9= 3. Вершина 5. Ребро 7-5.

1. k = 10. Вершина 8.

2. S_{10} = 1. Вершина 8. Ребро 5-8.

1. k = 11. Вершина 7.

2. S_{11} = 1. Вершина 7. Ребро 8-7.

1. $k = 12$. Вершина 6.

2. $S_{12} = 1$. Вершина 6. Ребро 7-6.

1. $k = 13$. Вершина 1.

2. $S_{13} = 1$. Вершина 1. Ребро 6-1.

1. $k = 14$. Вершин, связанных непомеченными ребрами, нет.

3. $S_{14} = 0$. В найденных циклах нет вершин, имеющих инцидентные непомеченные ребра.

4. Для нахождения ЭП необходимо объединить два найденных цикла 3-1-2-3 и 1-4-5-2-7-5-8-7-6-1 в вершине 1. Результатом объединения будет путь: 3-1-4-5-2-7-5-8-7-6-1-2-3.

5. Так как в этом пути нет вершин с непомеченными ребрами, то найденный путь является эйлеровым.

Определение связности графа

Неориентированный граф называется **связным**, если для любой пары вершин $x_i, x_j \in X$ существует хотя бы один маршрут, их соединяющий. В противном случае граф **несвязный**. Несвязный граф может быть единственным образом разбит на группы взаимосвязанных вершин (на **связные компоненты**) таким образом, что не существует маршрута, соединяющего какие-либо из вершин различных компонент.

Решим задачу *определения связности графа* с помощью элементов алгоритма Фаулкса.

Пусть дана матрица смежности R неориентированного графа G (матрица R симметрична относительно главной диагонали), имеющего $\boldsymbol{n}$ вершин. Необходимо определить, является ли этот граф связным. В случае, если граф несвязный, то требуется выделить из него все $\boldsymbol{m}$ связных подграфов $G_1, G_2, \ldots, G_m$ (связных компонент) таких, что, например, из любой вершины подграфа G_1 можно найти путь или маршрут, ведущий к любой другой вершине этого же подграфа, но нельзя найти путь или маршрут ни к одной вершине, не принадлежащей данному подграфу G_1. Требуется определить, какие вершины входят в каждый из подграфов.

На основании матрицы смежности R получим матрицу $R1$ достижимости не более чем за один шаг.

Затем, в соответствии с алгоритмом Фаулкса, найдем матрицу R^N путем булевского перемножения матриц ($R^N = R^{N-1} \otimes R^1$) до тех пор, пока не достигнем $R^N = R^{N-1}$. Полученная матрица R^N является матрицей достижимости графа G. Она позволяет выявить связные подграфы и номера вершин, входящих в связные подграфы. Для этого необходимо в некоторой i - строке матрицы R^N найти элементы $r^N_{ij} = 1$, $i, j = 1,\ldots,n$.

Номера столбцов j, где r_{ij} = 1, и будут номерами вершин, входящих в связный подграф, включающий в себя i-ю вершину. Рассматривая последовательно все несовпадающие строки матрицы можно выявить все ***m*** связных подграфов исходного графа G.

В случае ориентированного графа, для решения задачи выделения связных подграфов, не связных между собой путем или маршрутом, необходимо перейти от ориентированного графа к неориентированному, заменив дуги неориентированными ребрами (звеньями). При возникновении параллельных звеньев оставить для дальнейшего рассмотрения только по одному звену, связывающему соответствующие вершины. В остальном процесс поиска связных подграфов не меняется.

Пример.

Пусть дана матрица смежности R неориентированного графа G , имеющего n = 8 вершин. На основании матрицы R получим матрицу R^1 достижимости не более чем за один шаг :

$$R^1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Необходимо проверить граф G на связность и выявить все связные подграфы.

1. Найдем матрицу достижимости графа G путем последовательного булевого умножения $\otimes$ матрицы R^1 саму на себя до тех пор, пока не будет выполняться условие $R^N = R^{N\text{-}1}$. Полученная таким образом матрица R^N есть матрица достижимости.

В нашем случае на третьем шаге выполняется условие $R3 = R2$. Это означает, что путь длиной 2 - максимальный в данной графе, и дальнейшее перемножение матрицы не имеет смысла, так как мы будем получать одну и ту же матрицу. Найденная матрица R^3 есть матрица достижимости исходного графа. Матрица достижимости будет равна:

$$R^3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Наличие нулей в полученной матрице показывает, что между соответствующими элементами отсутствует путь любой длины и, следовательно, данный граф — несвязный.

2. Выбираем начальную вершину для выявления связных подграфов. Пусть i =1.

3. В строке $i=1$ выбираем все элементы $r^3{}_{1j}$, равные единице ($j=1,\ldots,n$). $r^3{}_{11} = r^3{}_{12} = r^3{}_{17} = r^3{}_{18} = 1$. Номера столбцов, в которых $r^3{}_{ij} = 1$ - это номера вершин первого связного подграфа, входящего в данный граф. То есть вершины 1, 2, 7, 8 образуют связный подграф G_1.

4. Определяем номер очередной вершины i, не входящей в связный подграф G_1 Пусть $i=3$. В строке 3 выбираем элементы, равные единице: $r^3{}_{33} = r^3{}_{36} = 1$. Следовательно, во второй связный подграф G_2 входят вершины 3, 6.

5. Определяем номер очередной вершины i, не входящей в связные подграфы G_1 и G_2. Пусть $i=4$. В строке 4 выбираем элементы, равные единице. $r^3{}_{44} = r^3{}_{45} = 1$. Следовательно, в третий связный подграф G_3 входят вершины 4, 5.

6. Так как проверены все вершины, то все связные подграфы найдены. Конец.

Таким образом, в результате работы алгоритма найдены три связных подграфа:

$G_1 = \{1, 2, 7, 8\}$; $G_2 = \{3, 6\}$; $G_3 = \{4, 5\}$ (см. рисунок)

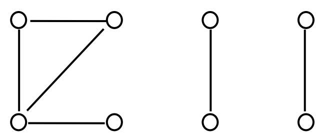


Рис. Исходный граф G..

Следует заметить, что при построении рисунка графа необходимо проверить наличие ребер в каждой сильной компоненте в соответствии с исходной матрицей смежности.

ЛИТЕРАТУРА

1. Босин П.Л., Булыгии В.С., Кулешов К.Г. Лабораторные работы но курсу «Основы теории конечных динамических систем», — М.: МАИ, 1985.
2. Булыгин В.С. Логические основы теории дискретных устройств: учеб. пособие. - М.: МАИ, 1983.
3. Булыгин В.С. Основы проектирования дискретных устройств АСУ: учеб. пособие. - М.: МАИ, 1982.
4. Просветов Г.И. Дискретная математика: задачи и решения: учебное пособие. – М.: БИНОМ. Лаборатория знаний,2008. – 222 с.
5. Кристофидес Н, Теория графов: Алгоритмический подход. - М.: Мир, 1978.
6. Булыгин В.С., Ескин В.И. Лабораторные работы «Дискретная математика (логические функции, конечные автоматы, графы)», — М.: МАИ, 2011.