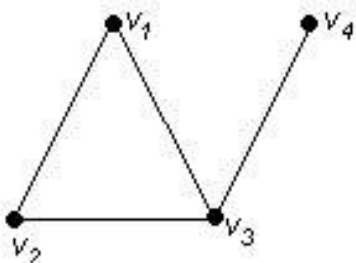


# Представление графов

## Матрица смежности

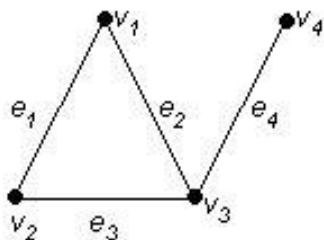


$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Достоинства – простота и наглядность, проверка наличия ребра между вершинами за один шаг (сложность  $O(1)$ ).

Недостатки – большой расход памяти.

## Матрица инцидентности

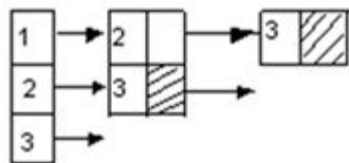
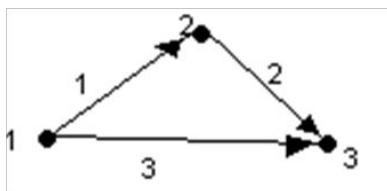


$$B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Достоинства – простота «стягивания» графа (сложность  $O(\log(n))$ ).

Недостатки – большой расход памяти, неочевидность некоторых алгоритмов.

## Списки смежности



Достоинства – низкий расход памяти.

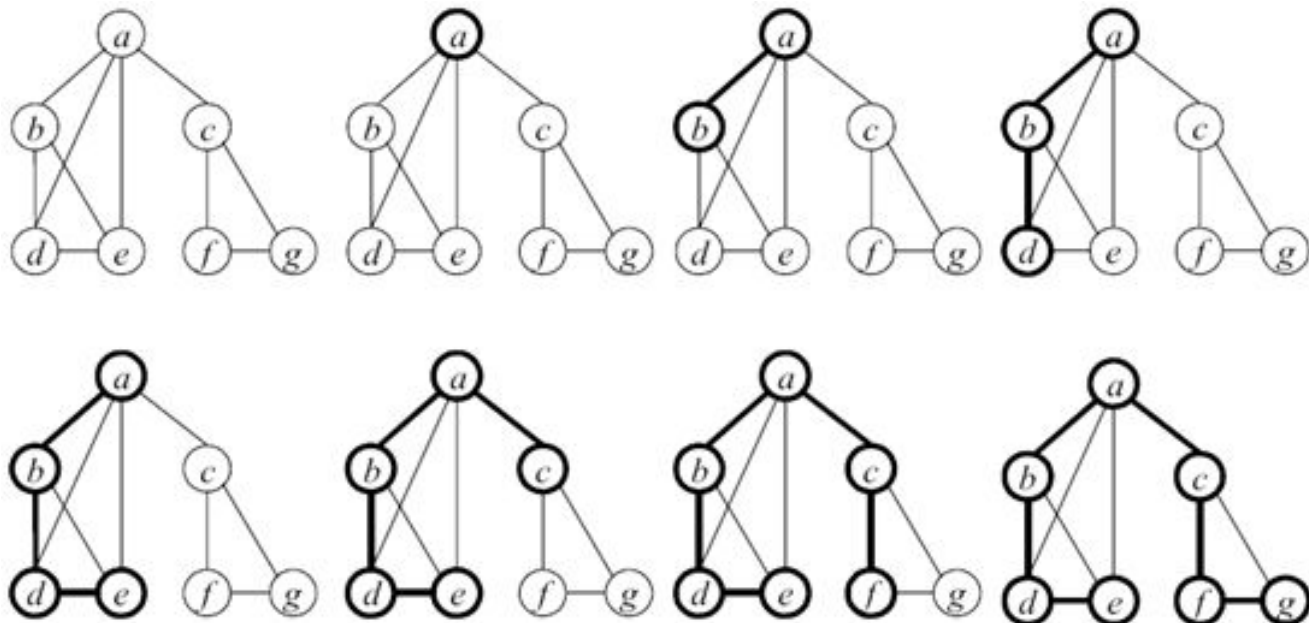
Недостатки – нет быстрого способа проверить существует ли ребро.

# Обход графа

Большое число задач, связанных с графами требует перебора вершин графа, т.е. просмотра каждой вершины в точности один раз (задачи поиска).

**Обход в глубину** осуществляется с некоторой вершины вдоль ребер графа, до попадания в лист. После этого нужно возвращаться назад вдоль пройденного пути, пока не будет обнаружена вершина, у которой есть еще не посещенная смежная вершина, и затем двигаться в направлении не посещённой вершины. Эти действия повторяются до возврата в начальную вершину после посещения всех остальных.

Т.о. основная идея поиска в глубину – сначала полностью исследовать одну ветку вглубь и только потом переходить к другим веткам.



Применения алгоритма поиска в глубину:

- поиск любого пути в графе;
- проверка, является ли одна вершина дерева предком другой;
- поиск наименьшего общего предка;
- топологическая сортировка (перенумеровать вершины графа таким образом, чтобы каждое рёбро вело из вершины с меньшим номером в вершину с большим);
- поиск компонент связности.

Одно из практических применений - проход по лабиринту с одним шагом видимости.

«Правило левой руки» (идти, ведя левой рукой по стенке) будет поиском в глубину, если лабиринт древовидный (нет замкнутых путей).

## Алгоритм поиска в глубину

**Вход:**  $G=(V, A)$  - неориентированный граф, представленный списками смежностей: для каждой вершины  $v$  список  $L_v$  содержит перечень всех смежных с  $v$  вершин.

**Выход:**  $NUM[v]$  - массив с номерами вершин в порядке их прохождения.

## Алгоритм ПОГ

1.  $NOMER = 1$ ;
2. для всех  $v$  положим  $NUM[v] = 0$  - пометим как "новую";
3. **ПОКА** существует "новая" вершина  $v$
4. **ВЫПОЛНЯТЬ**  $ПОИСК(v)$ .

Рекурсивная процедура поиска.

## Алгоритм $ПОИСК(v)$ :

1. пометить  $v$  как "старую"  $NUM[v] = NOMER$ ;
2.  $NOMER = NOMER + 1$ ;
3. **ДЛЯ КАЖДОЙ**  $w$  принадлежащей  $L_v$  **ВЫПОЛНЯТЬ**
4.   **ЕСЛИ** вершина  $w$  "новая"
5.   **ТО**
6.   {
7.      $ПОИСК(w)$ ;
8.   }

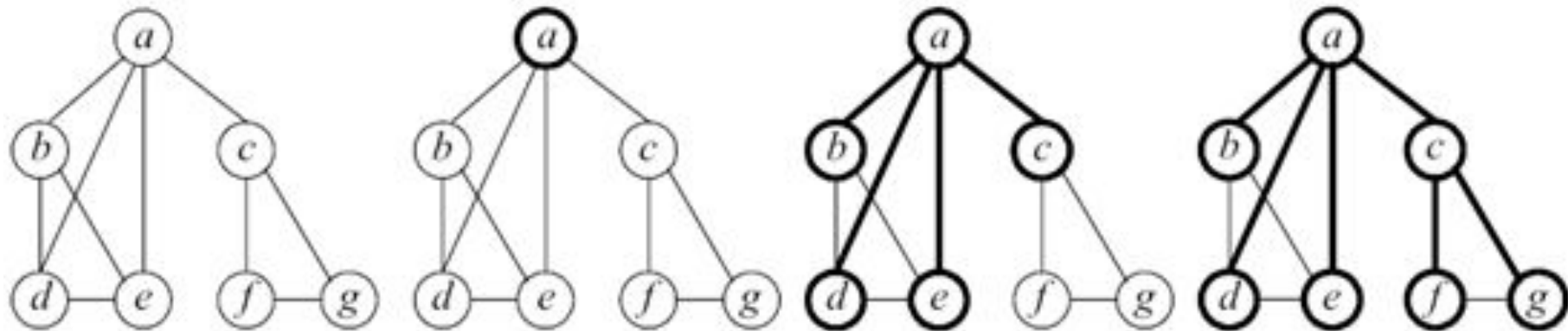
**Обход в ширину** осуществляется с некоторой вершины и после посещения первой вершины, посещаются все соседние с ней вершины.

Потом посещаются все вершины, находящиеся на расстоянии двух ребер от начальной.

При каждом новом шаге посещаются вершины, расстояние от которых до начальной на единицу больше предыдущего.

Чтобы предотвратить повторное посещение вершин, ведется список посещенных вершин.

**Обход в ширину** осуществляется по уровням высоты графа.



## Применения алгоритма поиска в ширину:

- трассировка печатных плат;
- поиск компонент связности в графе;
- поиск кратчайшего пути между двумя узлами не взвешенного графа;
- нахождение кратчайшего цикла в ориентированном не взвешенном графе.

### Алгоритм поиска в ширину

**Вход:**  $G=(V, A)$  - неориентированный граф, представленный списками смежностей: для каждой вершины  $v$  список  $L_v$  содержит перечень всех смежных с  $v$  вершин.

**Выход:**  $NUM[v]$  - массив с номерами вершин в порядке их прохождения.

### Алгоритм ПОШ

1. Создать пустую очередь  $Q = \{\}$  и  $NOMER = 1$  номер порядка прохождения;
2. Пометить  $v$  как посещённую  $NUM[v] = NOMER$ ;
3. Поместить  $v$  в очередь  $Q += v$ ;
4. **ПОКА**  $Q \neq \emptyset$
5. Удалить начальный элемент  $u$  из очереди  $Q$
6. **ДЛЯ КАЖДОЙ**  $w$  принадлежащей  $L_u$  **ВЫПОЛНЯТЬ**
7. Пометить  $w$  как посещённую  $NUM[w] = NOMER$ ;
8.  $NOMER = NOMER + 1$ ;
9. Поместить  $w$  в очередь  $Q += w$ .



[https://docs.google.com/forms/d/e/1FAIpQLSfqeQCVWYoQPj9CXc-M9sxhe0YLP8aMUAAe0242ZeuWLTMTpw/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfqeQCVWYoQPj9CXc-M9sxhe0YLP8aMUAAe0242ZeuWLTMTpw/viewform?usp=sf_link)

## Операции над графами

Унарными называются операции, определённые на одном графе (один граф в качестве операнда).

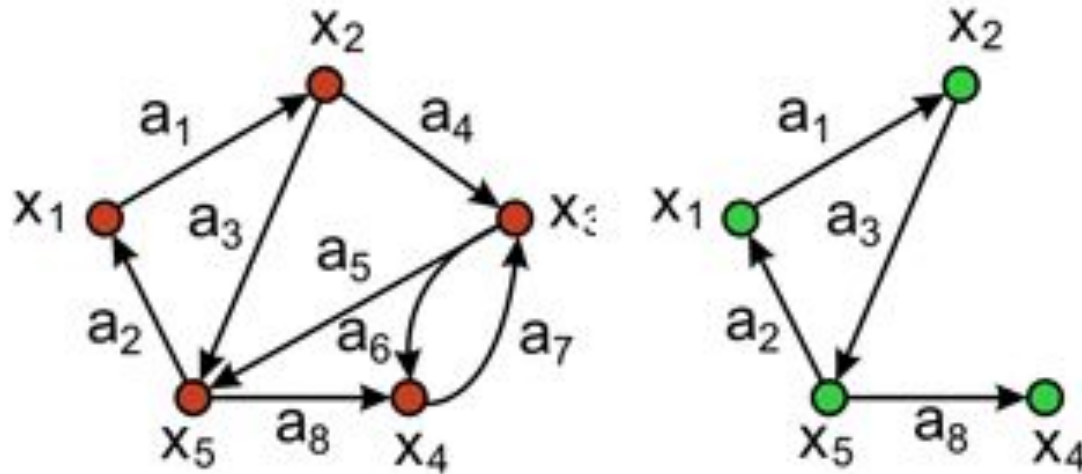
**Удаление вершины.** Операция порождает из графа  $G = (X, A)$ , граф  $G_1 = (X_1, A_1)$ , в котором  $X_1 = X - x_i$  и  $A_1 = A - a$ , где  $x_i$  - удаляемая вершина графа  $G$ ,  $a$  – множество инцидентных вершине  $x_i$  ребер.

Т.е.  $G_1$  получается путём удаления из графа  $G$  вершины  $x_i$  и всех ребер, инцидентных этой вершине.

Результирующая матрица смежности *графа* после выполнения операции удаления вершины  $x_i$  получается путем удаления соответствующего  $i$  - го столбца и  $i$  -ой строки из исходной матрицы .



# Пример удаления вершины



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	1				
$x_2$			1	1	
$x_3$				1	1
$x_4$			1		
$x_5$	1			1	

	$x_1$	$x_2$	$x_4$	$x_5$
$x_1$		1		
$x_2$				1
$x_4$				
$x_5$	1		1	

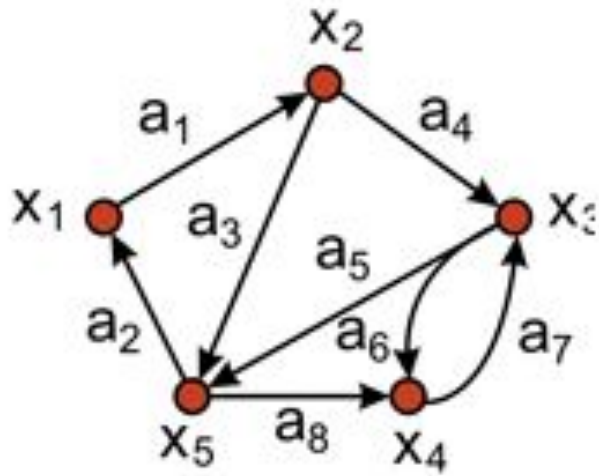
## Операции над графами

**Удаление ребра или дуги.** Если  $a_i$  - дуга графа  $G = (X, A)$ , то операция удаления  $a_i$  порождает новый граф  $G_1 = (X, A_1)$ , в котором  $A_1 = A - a_i$ .

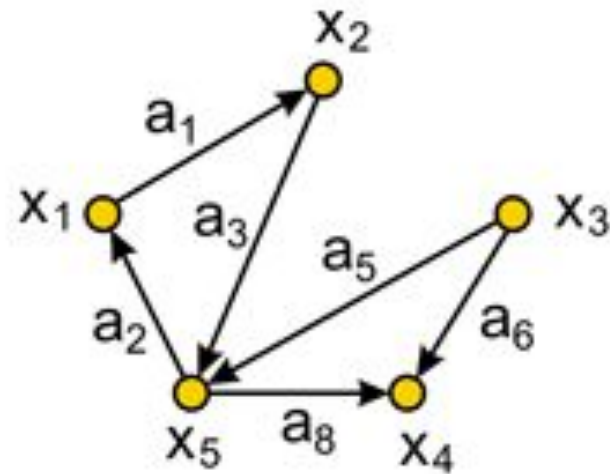
**!!! Концевые вершины дуги  $a_i$  при этом не удаляются !!!**

Результирующая матрица смежности графа  $G_1$  после выполнения операции удаления дуги  $a_i$  получается путем удаления соответствующих элементов из исходной матрицы смежности графа  $G$ .

# Пример удаления ребра



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	1				
$x_2$			1	1	
$x_3$				1	1
$x_4$			1		
$x_5$	1			1	



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	1				
$x_2$				1	
$x_3$				1	1
$x_4$			1		
$x_5$	1			1	

## Операции над графами

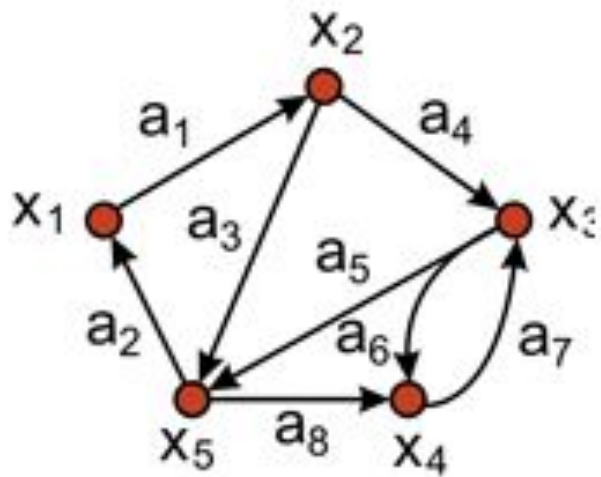
### ***Замыкание или отождествление.***

Пара вершин  $x_i$  и  $x_j$  в *графе*  $G$  замыкается (или отождествляется), если они заменяются такой новой вершиной, что все дуги в *графе*  $G$ , инцидентные  $x_i$  и  $x_j$ , становятся инцидентными новой вершине.

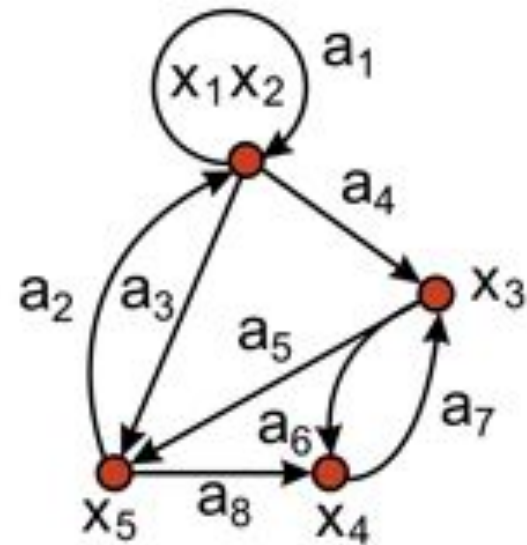
**!!! При этом ребро, соединяющее вершины  $x_i$  и  $x_j$  становится петлёй !!!**

Матрица смежности графа после выполнения операции замыкания вершин  $x_i$  и  $x_j$  получается путем поэлементного логического сложения  $i$  - го и  $j$  - го столбцов и  $i$  -ой и  $j$  - строк в исходной матрице.

# Пример замыкания вершин



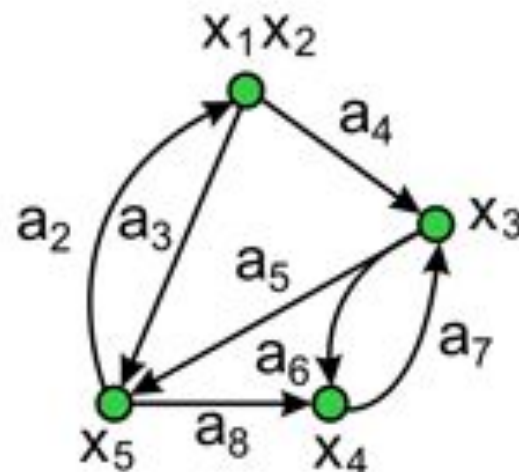
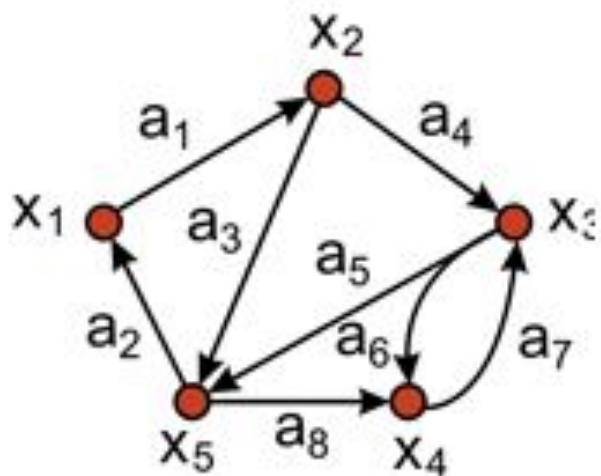
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	1				
$x_2$			1	1	
$x_3$				1	1
$x_4$			1		
$x_5$	1			1	



	$x_{1-2}$	$x_3$	$x_4$	$x_5$
$x_{1-2}$	1	1		1
$x_3$			1	1
$x_4$		1		
$x_5$	1		1	

## Стягивание.

Под стягиванием подразумевают операцию удаления дуги или ребра и отождествление его концевых вершин.



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	1				
$x_2$			1	1	
$x_3$				1	1
$x_4$			1		
$x_5$	1			1	

	$x_{1-2}$	$x_3$	$x_4$	$x_5$
$x_{1-2}$		1		1
$x_3$			1	1
$x_4$		1		
$x_5$	1			1

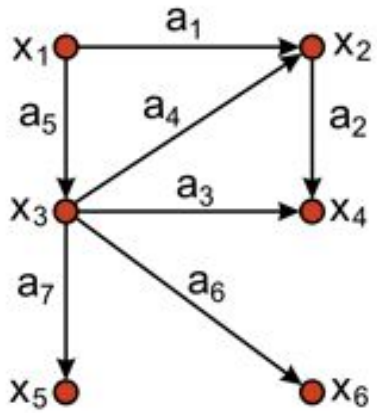
## Операции над графами

Бинарными называются операции, определённые на двух графах (два графа в качестве операндов).

**Объединение** графов  $G_1$  и  $G_2$ , обозначаемое как  $G_1 \cup G_2$ , представляет собой *граф*  $G_3 = (X_1 \cup X_2, A_1 \cup A_2)$  такой, что множество его вершин является объединением  $X_1$  и  $X_2$ , а множество ребер – объединением  $A_1$  и  $A_2$  .

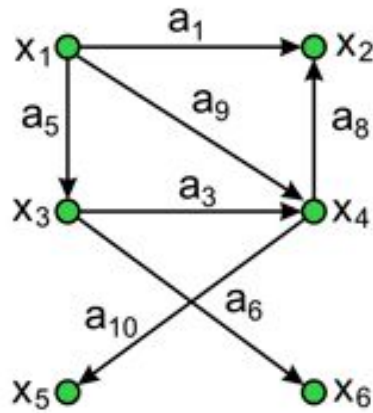
Матрица смежности результирующего графа  $G_3$  получается операцией поэлементного логического сложения матриц смежности исходных графов  $G_1$  и  $G_2$  .

# Пример объединения графов



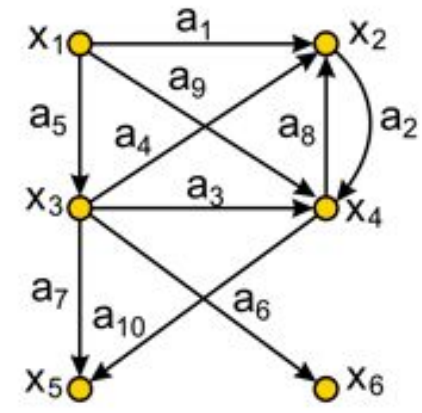
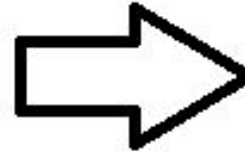
$$A_1 =$$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$		1	1			
$x_2$				1		
$x_3$		1		1	1	1
$x_4$						
$x_5$						
$x_6$						



$$A_2 =$$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$		1	1	1		
$x_2$						
$x_3$				1		1
$x_4$		1			1	
$x_5$						
$x_6$						



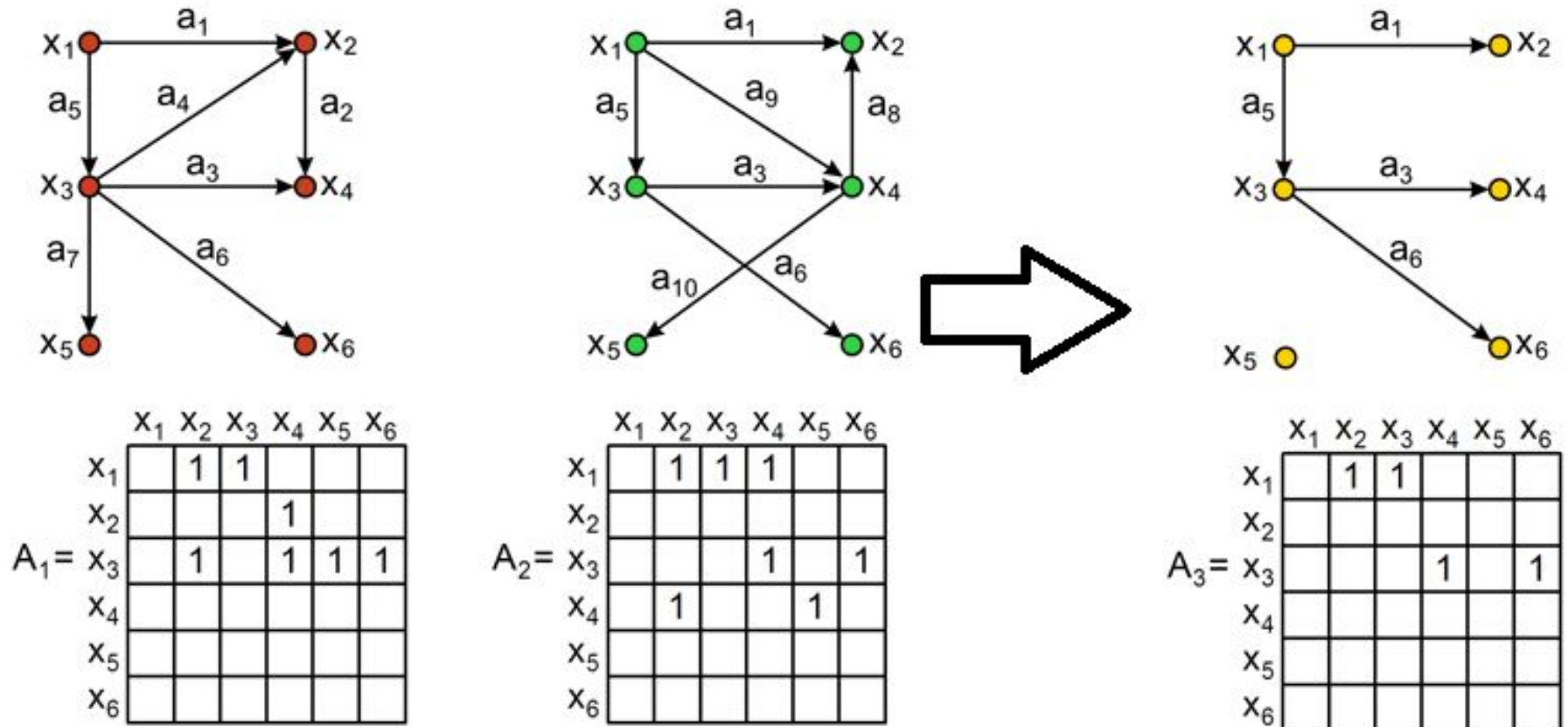
$$A_3 =$$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$		1	1	1		
$x_2$				1		
$x_3$		1		1	1	1
$x_4$		1			1	
$x_5$						
$x_6$						



**Пересечение графов**  $G1$  и  $G2$ , обозначаемое как  $G1 \cap G2$ , представляет собой **граф**  $G3 = (X1 \cap X2, A1 \cap A2)$  такой, что множество его вершин и дуг состоит из вершин и дуг, присутствующих одновременно в  $G1$  и  $G2$ .

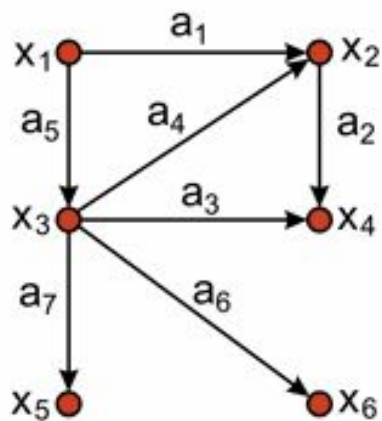
Результирующая матрица смежности получается операцией поэлементного логического умножения матриц смежности исходных **графов**  $G1$  и  $G2$ .

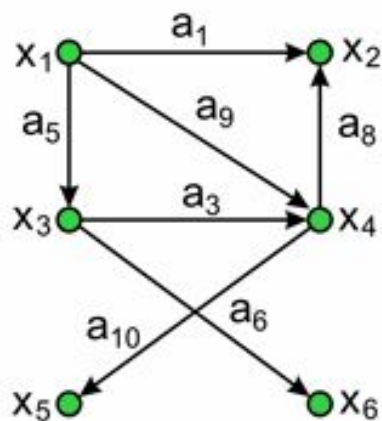


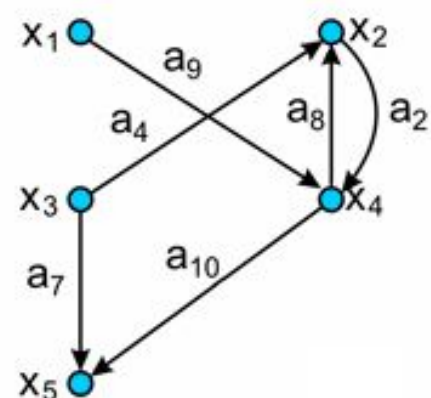
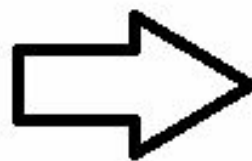
**Кольцевая сумма графов  $G_1$  и  $G_2$** , обозначаемая как  $G_1 \oplus G_2$ , представляет собой *граф*, порожденный на множестве ребер  $A_1 \oplus A_2$ .

Т.е. новый *граф* не имеет изолированных вершин и состоит только из ребер, присутствующих либо в  $G_1$ , либо в  $G_2$ , но не в обоих одновременно.

Результирующая матрица смежности получается операцией поэлементного логического сложения по mod 2 матриц смежности исходных *графов*  $G_1$  и  $G_2$ .



$$A_1 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} & 1 & 1 & & & \\ & & & 1 & & \\ 1 & & 1 & 1 & 1 & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix} \end{matrix}$$


$$A_2 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} & 1 & 1 & 1 & & \\ & & & & & \\ & & & 1 & & 1 \\ & 1 & & & 1 & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix} \end{matrix}$$


$$A_3 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} & & & 1 & \\ & & & 1 & \\ 1 & & & 1 & \\ & 1 & & & 1 \\ & 1 & & & 1 \end{bmatrix} \end{matrix}$$

**Декартово произведение графов**  $G_1 = (X, A)$  и  $G_2 = (Y, B)$ , обозначаемое как  $G_1 \times G_2$ , представляет собой *граф*, множество вершин которого является декартовым *произведением* множеств  $X \times Y$ , т.е. вершины этого графа – все упорядоченные пары  $(x, y)$ . При этом вершины  $(x_1, y_1)$  и  $(x_2, y_2)$  в новом графе смежны, если  $x_1 = x_2$  и  $y_1$  смежна с  $y_2$  в графе  $G_2$ , или  $y_1 = y_2$  и  $x_1$  смежна с  $x_2$  в графе  $G_1$ .

