Лабораторная работа №2

НАХОЖДЕНИЕ КРАТЧАЙШЕГО РАССТОЯНИЯ НА СЕТИ С ПОМОЩЬЮ АЛГОРИТМА ФЛОЙДА

Алгоритм Флойда является одним из популярных алгоритмов маршрутизации, позволяющий находить кратчайший путь на сети связи, заменяя прямой маршрут между двумя узлами на маршрут с одним промежуточным пунктом. Рассмотрим как это реализуется на практике. Пусть есть три узла i, j и k и заданы расстояния между ними (рис. 1). Если выполняется неравенство $d_{ij} + d_{jk} < d_{ik}$, то целесообразно заменить путь $i \rightarrow k$ путем $i \rightarrow j \rightarrow k$. Такая замена (далее ее будем условно называть треугольным оператором) выполняется систематически в процессе выполнения алгоритма Флойда. Алгоритм Флойда требует выполнения нескольких операций.

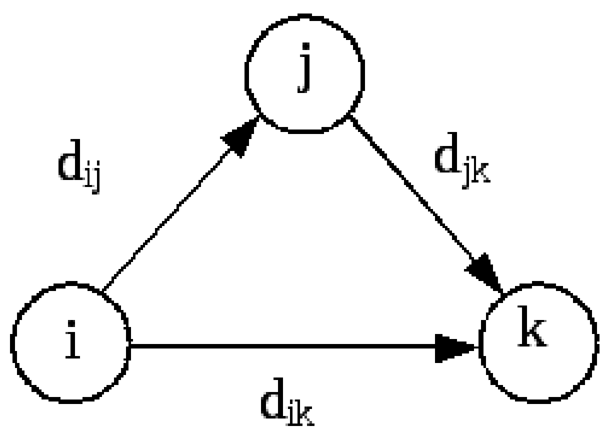


Рис. 1. Треугольный оператор Флойда.

Шаг 0. На этом шаге определим начальную матрицу расстояний D_0 и матрицу последовательности узлов S_0. Диагональные элементы обеих матриц помечаются знаком "-", показывающим, что эти элементы в вычислениях не участвуют. Полагаем $k = 1$, тогда начальная ситуация будет выглядеть следующим образом:

$D_0=$		1	2	…	j	…	n
	1	-	d_{12}	…	d_{1j}	…	d_{1n}
	2	d_{21}	-	…	d_{2j}	…	d_{2n}
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	i	d_{i1}	d_{i2}	…	-	…	d_{in}
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	n	d_{n1}	d_{n2}		d_{nj}		-

$S_0=$		1	2	…	j	…	n
	1	-	2	…	j	…	n
	2	1	-	…	j	…	n
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	i	1	2	…	-	…	n
	. .	. .	. .	. .	. .	. .	. .
	n	1	2		j		-

Основной шаг k. Задаем строку k и столбец k как ведущую строку и ведущий столбец. Рассматриваем возможность применения треугольного оператора ко всем элементам d_{ij} матрицы D_{k-1}. Если выполняется неравенство $d_{ik} + d_{kj} < d_{ij}$, тогда выполняем следующие действия:

- создаем матрицу D_k путем замены в матрице Dk-1 элемента dij на сумму $dik + dkj$,

- создаем матрицу S_k путем замены в матрице S_{k-1} элемента s_{ij} на k. Полагаем $k = k + 1$ и повторяем шаг k.

Поясним действия, выполняемые на k-м шаге алгоритма, представив матрицу D_{k-1} так, как она показана на рисунке 2. На этом рисунке строка k и столбец k являются ведущими. Строка i - любая строка с номером от 1 до k - 1, а строка p - произвольная строка с номером от $k + 1$ до n. Аналогично столбец j представляет любой столбец с номером от 1 до $k - 1$, столбец q - произвольный столбец с номером от $k + 1$ до n. Треугольный оператор выполняется следующим образом. Если сумма элементов ведущих строки и столбца (показанных в квадратах) меньше элементов, находящихся в пересечении столбца и строки (показанных в кружках), соответствующих рассматриваемым ведущим элементам, то расстояние (элемент в кружке) заменяется на сумму расстояний, представленных ведущими элементами:

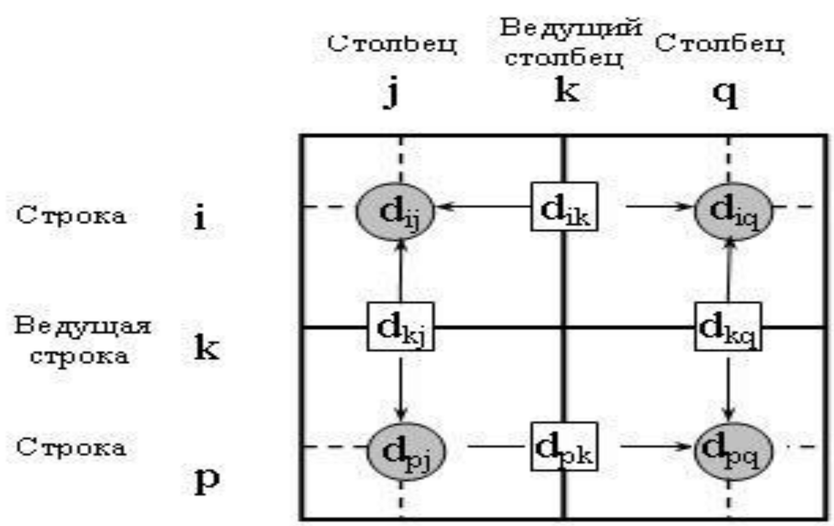


Рис. 2. Иллюстрация алгоритма Флойда

После реализации n шагов алгоритма определение по матрицам D_n и S_n кратчайшего пути между узлами i и j выполняется по следующим правилам.

1. Расстояние между узлами i и j равно элементу d_{ij} в матрице D_n.
2. Промежуточные узлы пути от узла i к узлу j определяем по матрице S_n. Пусть $s_{ij} = k$, тогда имеем путь $i \to k \to j$. Если далее $s_{ik} = k$ и $s_{kj} = j$, тогда считаем, что весь путь определен, так как найдены все промежуточные узлы. В противном случае повторяем описанную процедуру для путей от узла i к узлу k и от узла k к узлу j.

Теперь рассмотрим реализацию алгоритма на примере.

Пусть для сети, показанной на рисунке 3, требуется найти кратчайшие пути между любыми двумя узлами. Расстояние между узлами этой сети проставлены на рисунке возле соответствующих ребер. Ребро *(3, 5)* ориентированно, поэтому не допускается движение от узла *5* к узлу *3*. Все остальные ребра позволяют движение в обе стороны:

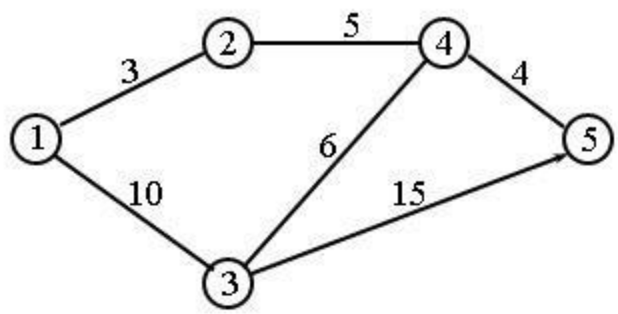


Рис. 3. Пример сети

<u>*Шаг 0*</u>. Начальные матрицы *D0* и *S0* строятся непосредственно по заданной схеме сети. Матрица *D0* симметрична, за исключением пары элементов *d35* и *d53*, где *d53* равно бесконечности, поскольку невозможен переход от узла *5* к узлу *3*:

D_0

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	∞	5	∞
3	10	∞	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_0

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	3	4	5
3	1	2	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Рис. 4. Начальное состояние

<u>*Шаг 1*</u>. В матрице D_0 выделены ведущие строка и столбец $(k = 1)$. Двойной рамкой представлены элементы d_{23} и d_{32}, единственные среди элементов матрицы D_0, значения которых можно улучшить с помощью треугольного оператора. Таким образом, чтобы на основе матриц D_0 и S_0 получить матрицы D_1 и S_1, выполняем следующие действия:

- Заменяем d_{23} на $d_{21} + d_{13} = 3 + 10 = 13$ и устанавливаем $s_{23} = 1$.
- Заменяем d_{32} на $d_{31} + d_{12} = 10 + 3 = 13$ и устанавливаем $s_{32} = 1$.

Матрицы D_1 и S_1 имеют следующий вид:

D_1

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	**13**	5	∞
3	10	**13**	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_1

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	**1**	4	5
3	1	**1**	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Рис. 5. Матрицы D_1 и S_1

<u>*Шаг 2.*</u> Полагаем $k = 2$; в матрице D_1 выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матрицы D_1 и S_1, выделенным двойной рамкой. В результате получаем матрицы D_2 и S_2:

D_2

	1	2	3	4	5
1	—	3	10	**8**	∞
2	3	—	13	5	∞
3	10	13	—	6	15
4	**8**	5	6	—	4
5	∞	∞	∞	4	—

S_2

	1	2	3	4	5
1	—	2	3	**2**	5
2	1	—	1	4	5
3	1	1	—	4	5
4	**2**	2	3	—	5
5	1	2	3	4	—

Рис. 6. Матрицы D_2 и S_2

<u>Шаг 3.</u> Полагаем $k = 3$; в матрице D_2 выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матрицы D_2 и S_2, выделенным двойной рамкой. В результате получаем матрицы D_3 и S_3:

D_3

	1	2	3	4	5
1	—	3	10	8	**25**
2	3	—	13	5	**28**
3	10	13	—	6	15
4	8	5	6	—	4
5	∞	∞	∞	4	—

S_3

	1	2	3	4	5
1	—	2	3	2	**3**
2	1	—	1	4	**3**
3	1	1	—	4	5
4	2	2	3	—	5
5	1	2	3	4	—

Рис. 7. Матрицы D_3 и S_3

<u>Шаг 4.</u> Полагаем $k = 4$, ведущие строка и столбец в матрице D_3 выделены. Получаем новые матрицы D_4 и S_4:

D_4

	1	2	3	4	5
1	—	3	10	8	**12**
2	3	—	**11**	5	**9**
3	10	**11**	—	6	**10**
4	8	5	6	—	4
5	**12**	**9**	**10**	4	—

S_4

	1	2	3	4	5
1	—	2	3	2	**4**
2	1	—	**4**	4	**4**
3	1	**4**	—	4	**4**
4	**2**	2	3	—	5
5	**4**	**4**	**4**	4	—

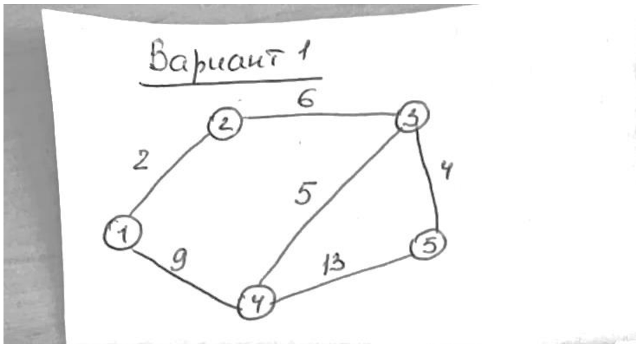
Рис. 8. Матрицы D_4 и S_4

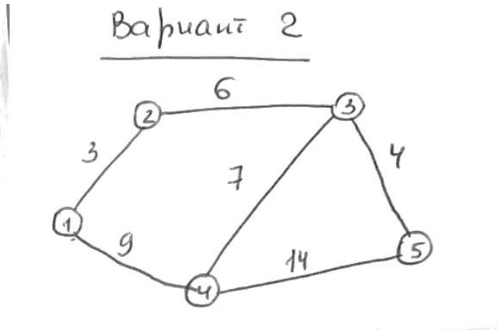
Шаг 5. Полагаем $k = 5$, ведущие строка и столбец в матрице D_4 выделены. Никаких действий на этом шаге не выполняем; вычисления закончены.

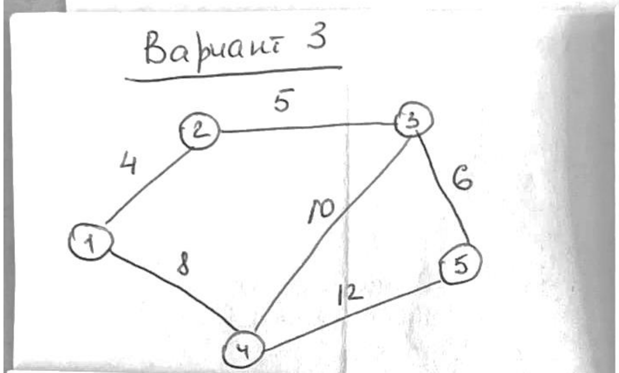
Конечные матрицы D_4 и S_4 содержат всю информацию, необходимую для определения кратчайших путей между любыми двумя узлами сети. Например, кратчайшее расстояние между узлами *1* и *5* равно $d_{15} = 12$.

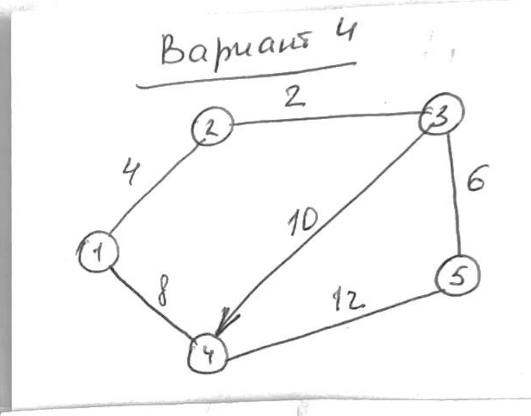
Для нахождения соответствующих маршрутов напомним, что сегмент маршрута (i, j) состоит из ребра (i, j) только в том случае, когда $s_{ij} = j$. В противном случае узлы i и j связаны, по крайней мере, через один промежуточный узел. Например, поскольку $s_{15} = 4$ и $s_{45} = 5$, сначала кратчайший маршрут между узлами *1* и *5* будет иметь вид $1 \to 4 \to 5$. Но так как s_{14} не равно *4*, узлы *1* и *4* в определенном пути не связаны одним ребром (но в исходной сети они могут быть связаны непосредственно). Далее следует определить промежуточный узел (узлы) между первым и четвертым узлами. Имеем $s_{14} = 2$ и $s_{24} = 4$, поэтому маршрут $1 \to 4$ заменяем $1 \to 2 \to 4$. Поскольку $s_{12} = 2$ и $s_{24} = 4$, других промежуточных узлов нет. Комбинируя определенные сегменты маршрута, окончательно получаем следующий кратчайший путь от узла *1* до узла *5*: $1 \to 2 \to 4 \to 5$. Длина этого пути равна *12* километрам.

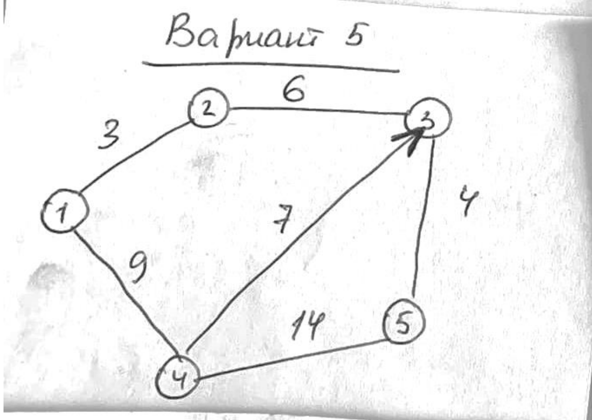
Исходные данные к работе

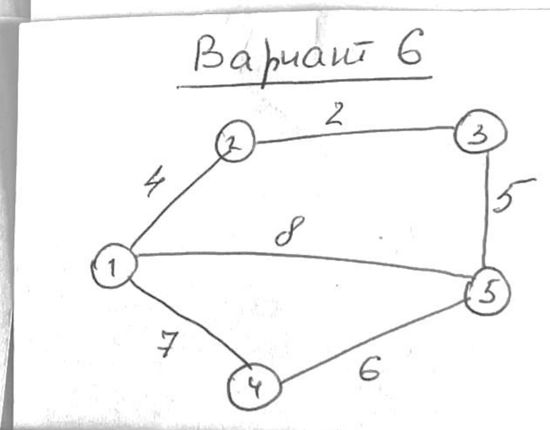


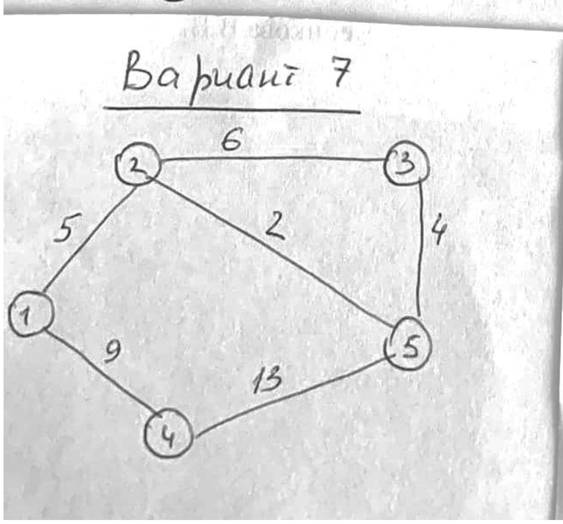


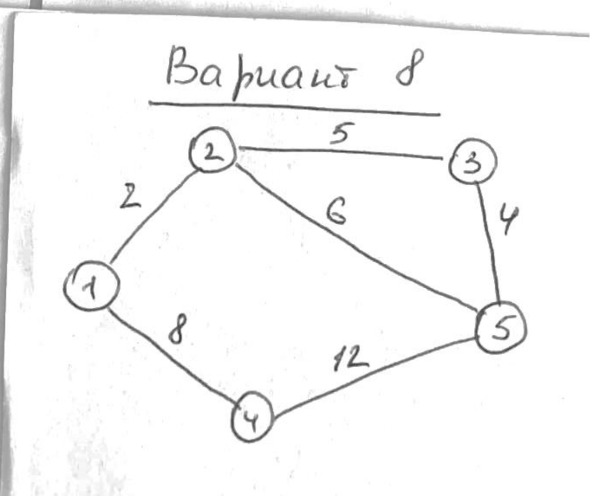




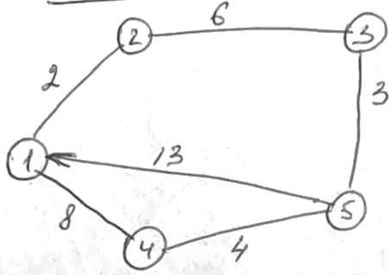




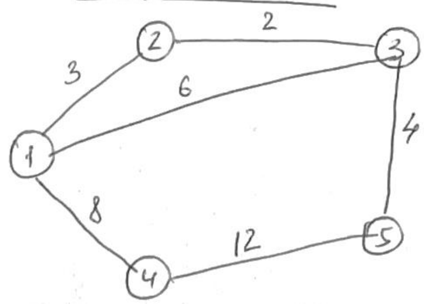




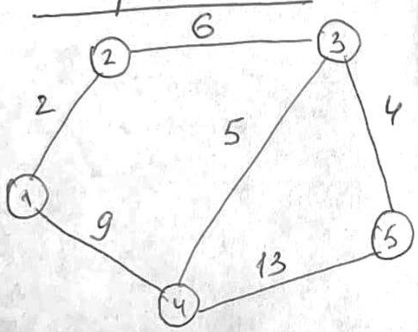
Вариант 9



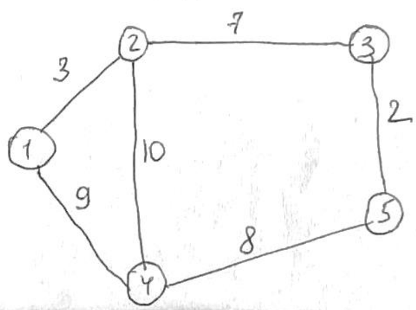
Вариант 10



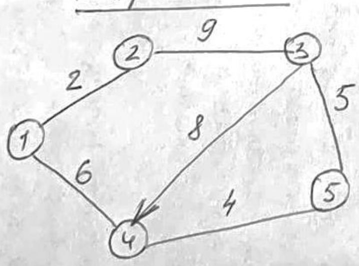
Вариант 11



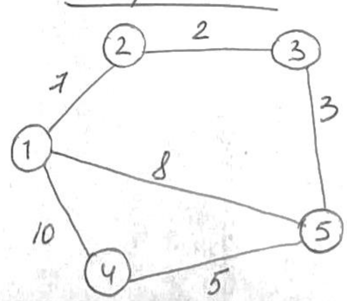
Вариант 12



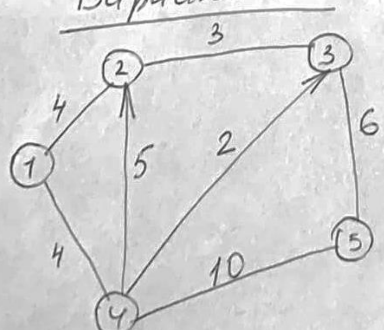
Вариант 13



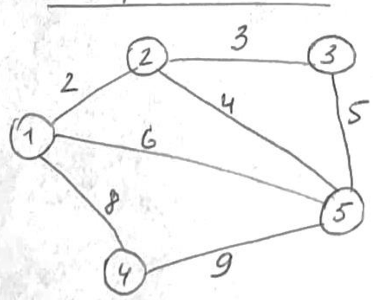
Вариант 14



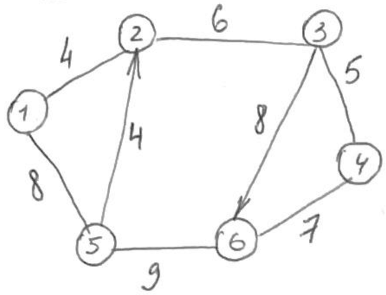
Вариант 15



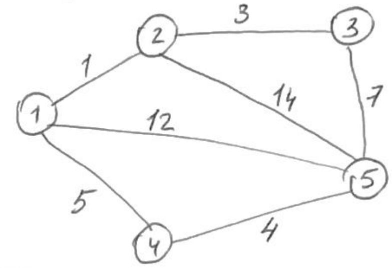
Вариант 16



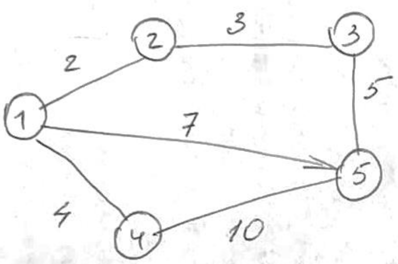
Вариант 17



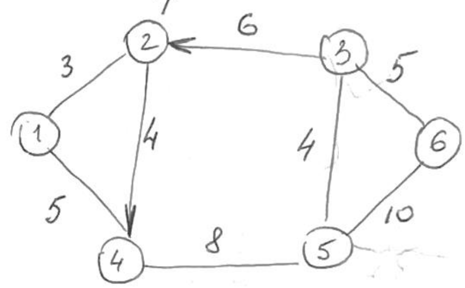
Вариант 18



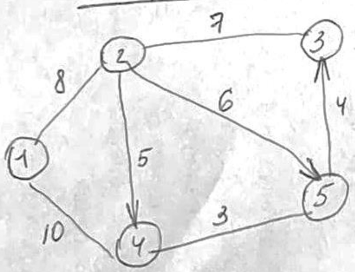
Вариант 19



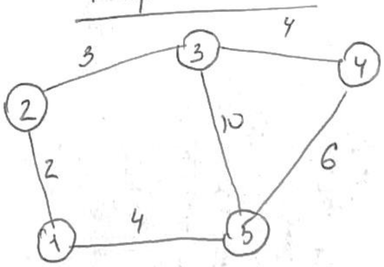
Вариант 20



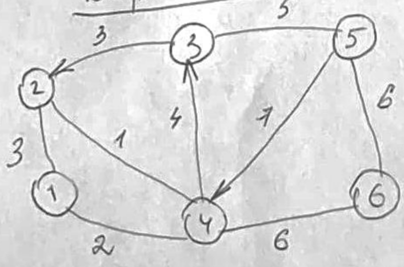
Вариант 22



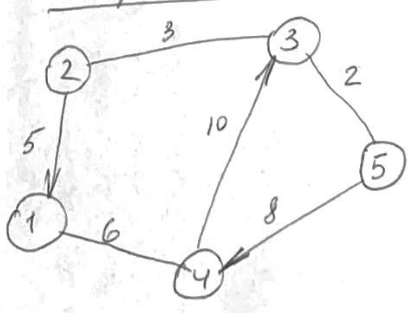
Вариант 23

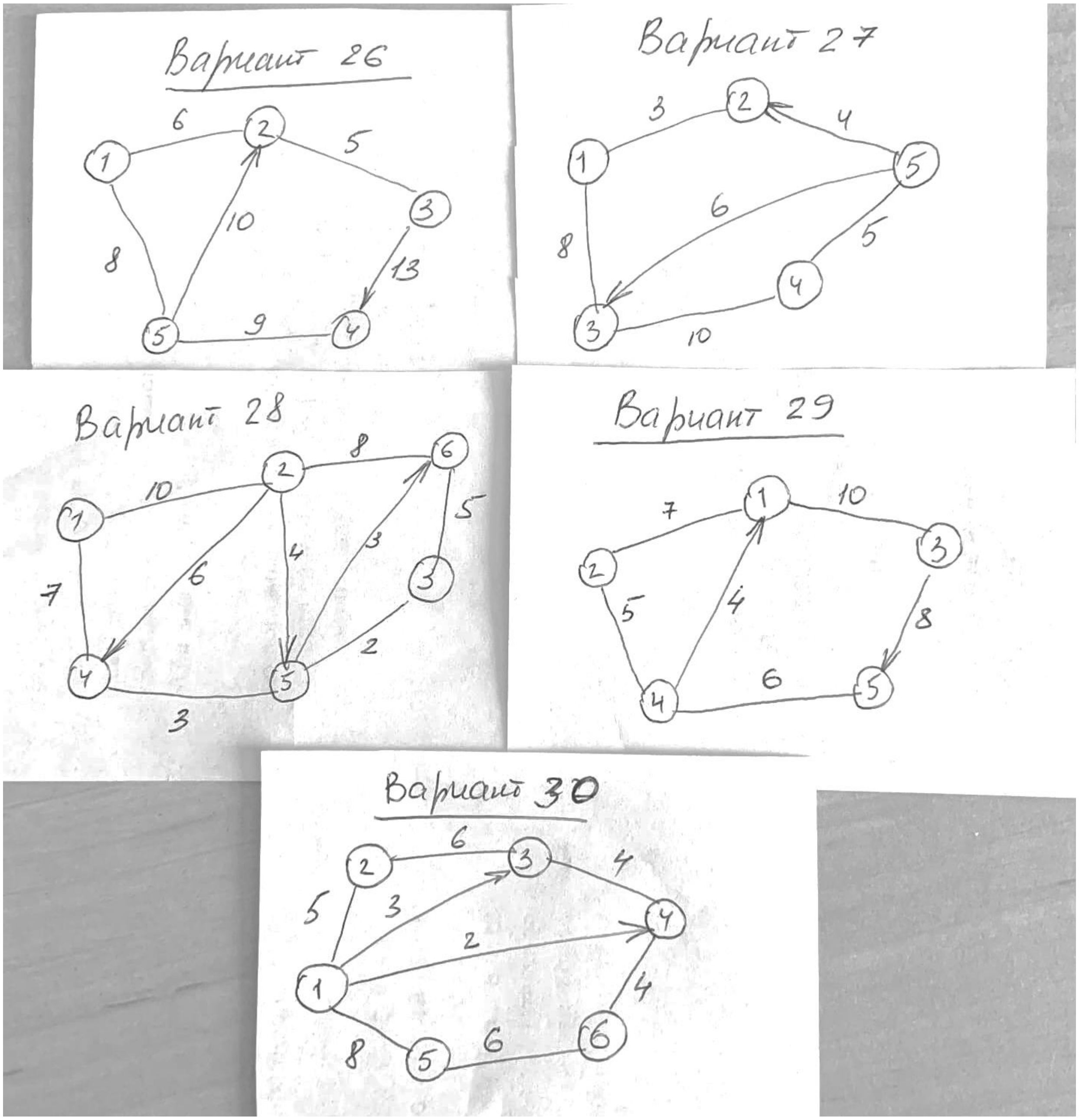


Вариант 24



Вариант 25





Порядок выполнения работы:

1. Получить исходные данные к работе.
2. Построить сеть связи.
3. Используя метод Флойда, определить кратчайшие расстояния в сети.

Содержание отчета:

1. Цель работы.
2. Ручной расчет сети.
3. Блок-схема алгоритма, расчет на ЭВМ.