Южный федеральный университет

Кафедра синергетики и процессов управления

Презентация

к индивидуальному заданию №1

«Сетевые модели в задачах принятия решения»

по дисциплине

«Системный анализ и принятие решений»

Направление подготовки 220100.62.01 «Системный анализ и управление»

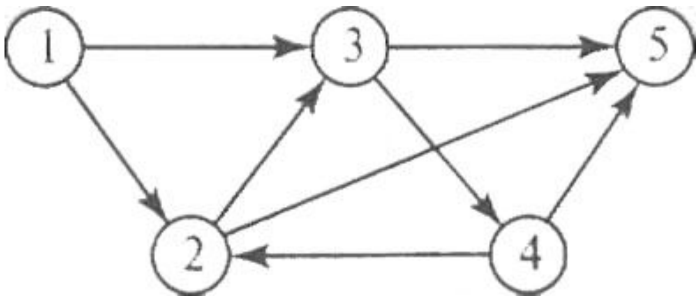
1.1 Построение минимального остовного дерева в задачах принятия решения

Не менее 70% реальных задач математического программирования, составляющих большинство задач системного анализа, можно представить в виде сетевых моделей.

Примеры типовых задач	**Способ решения задачи**
1. Проектирование газопровода, соединяющего буровые скважины морского базирования с находящейся на берегу приемной станцией. Целевая функция соответствующей модели должна минимизировать стоимость строительства газопровода.	Алгоритм нахождения минимального остовного дерева.
2. Минимизация кабельных сетей.	
3. Минимизация транспортных перевозок и др.	

1.1.1 Основные понятия и определения

Граф (сеть) состоит из множества узлов, связанных дугами (или ребрами), и описывается парой множеств (N, A), где N - множество узлов, а A - множество ребер. Например, сеть, показанная на рисунке, описывается следующим образом:



N = {1, 2, 3, 4, 5},

A = {(1, 2), (1, 3), (2, 3), (2, 5), (3, 4),
(3, 5), (4, 2), (4, 5)}.

С каждым типом сети связан определенный тип потоков (например, транспортный поток нефти в нефтепроводах или автомобильные потоки в сети городских дорог). В общем случае потоки в сети ограничены пропускной способностью ее ребер, которая может быть как *конечной*, так и *бесконечной*.

Ребро называется *направленным*, или *ориентированным* (и в этом случае ребро будем называть **дугой**), если в одном направлении возможен только положительный поток, а в противоположном - только нулевой.

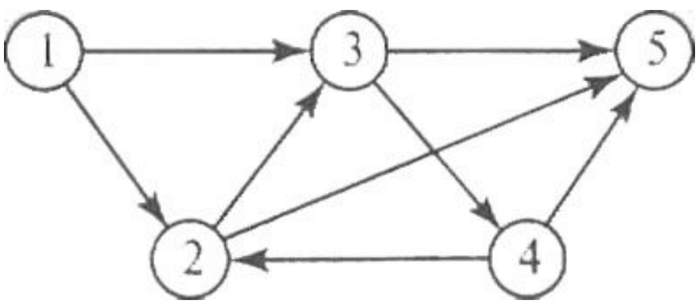
В **ориентированной сети** все ребра ориентированы.

Путем называется последовательность различных ребер, соединяющих два узла.

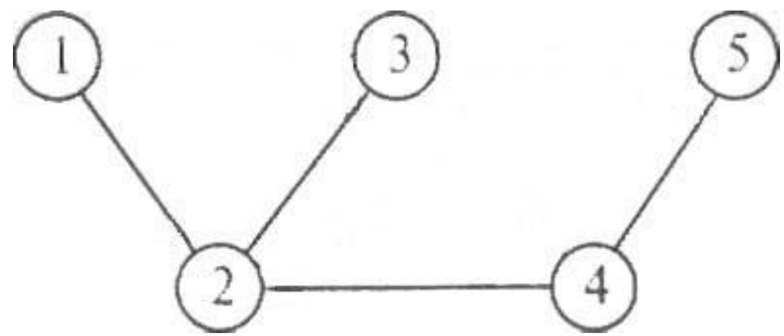
Путь формирует **цикл**, если начальный и конечный узлы совпадают.

Ориентированный цикл - это цикл, в котором все дуги ориентированы в определенном направлении.

***О*стовное дерево** - это дерево, содержащее *<u>все</u>* узлы сети. В сети из n узлов остовное дерево содержит n-1 ребер и возможно построить n^{n-2} таких деревьев.



Исходная сеть



*О*стовное дерево

1.1.2 Алгоритм построения минимального остовного дерева

Алгоритм построения минимального остовного дерева предполагает соединение **всех** узлов сети с помощью путей наименьшей длины.

Опишем процедуру выполнения этого алгоритма.
Обозначим через N = {1, 2,…, n} множество узлов сети и введем новые обозначения:
1) C_k - множество узлов сети, соединенных алгоритмом после выполнения *k*-й итерации этого алгоритма;

2) $\overline{C}_k$ - множество узлов сети, не соединенных с узлами множества C_k после выполнения *k*-й итерации этого алгоритма.

Этап 0. Пусть $C_0 = \varnothing$ и $\overline{C}_0 = N$. Полагаем *k* = 1.

Этап 1. Выбираем *любой* узел *i* из множества $\overline{C}_0$ и определяем $C_1 = \{i\}$, тогда $\overline{C}_1 = N - \{i\}$. Полагаем *k* = 2.

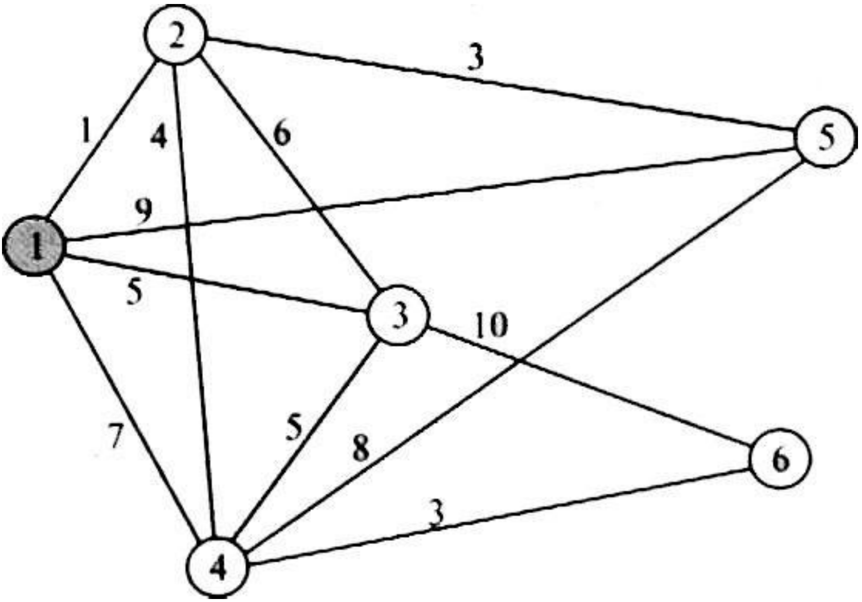
Основной этап *k*. В множестве $\overline{C}_{k-1}$, выбираем узел *j**, который соединен ***самой короткой дугой*** с каким-либо узлом из множества C_{k-1}. Узел *j** присоединяется к множеству C_{k-1} и удаляется из множества $\overline{C}_{k-1}$, т.е.

$$C_k = C_{k-1} + \{j^*\}, \quad \overline{C}_k = \overline{C}_{k-1} - \{j^*\}.$$

Если множество $\overline{C}_k$ пусто, то выполнение алгоритма заканчивается.

В противном случае полагаем *k* = *k* + 1 и повторяем основной этап.

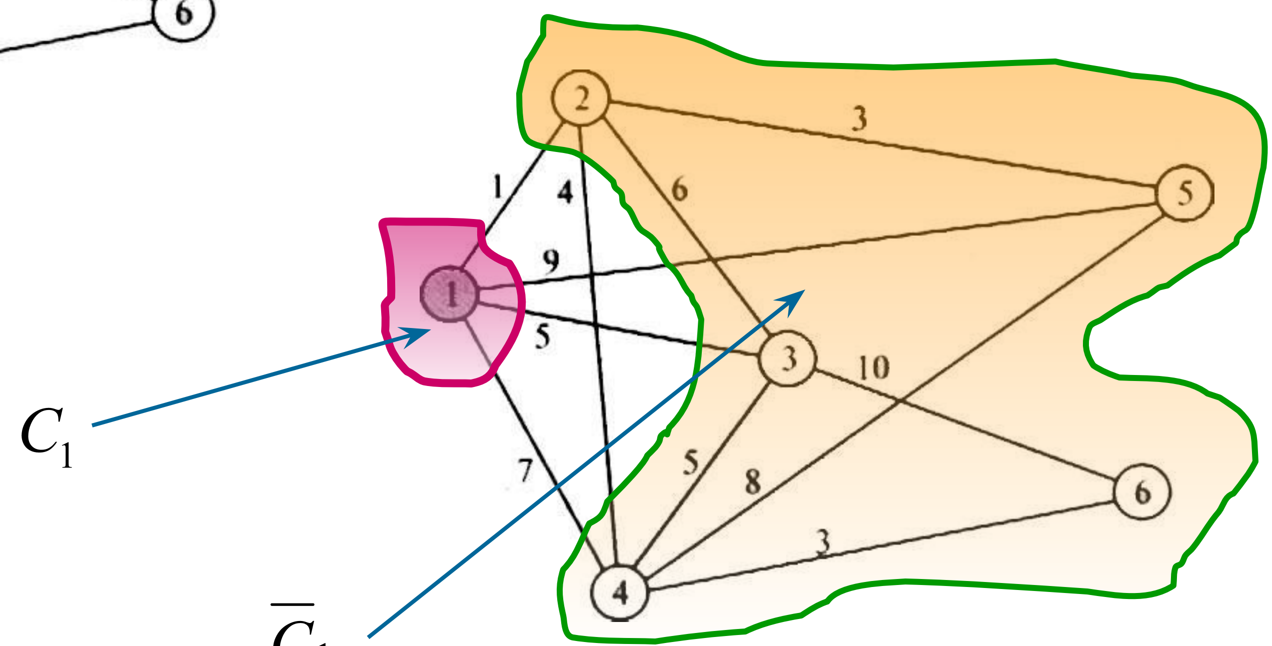
Пример 1. Телевизионная компания планирует подключение к своей кабельной сети пяти новых районов. На рисунке показана структура планируемой сети и расстояния (в км) между районами и телецентром – узел 1. Необходимо спланировать наиболее экономичную кабельную сеть – сеть минимальной длины.



Этап 0. $C_0 = \varnothing$ и $\overline{C}_0 = \{1,2,3,4,5,6\}$

Этап 1. Выбираем *i=1*, тогда $C_1 = \{1\}$,

$$\overline{C}_1 = \{2,3,4,5,6\}$$

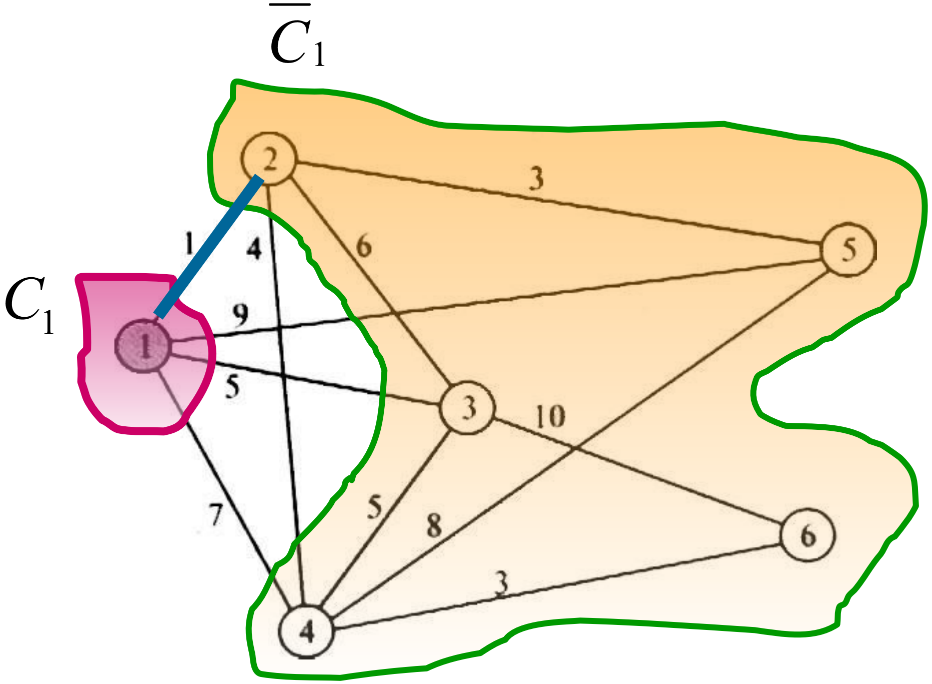


Итерация 1

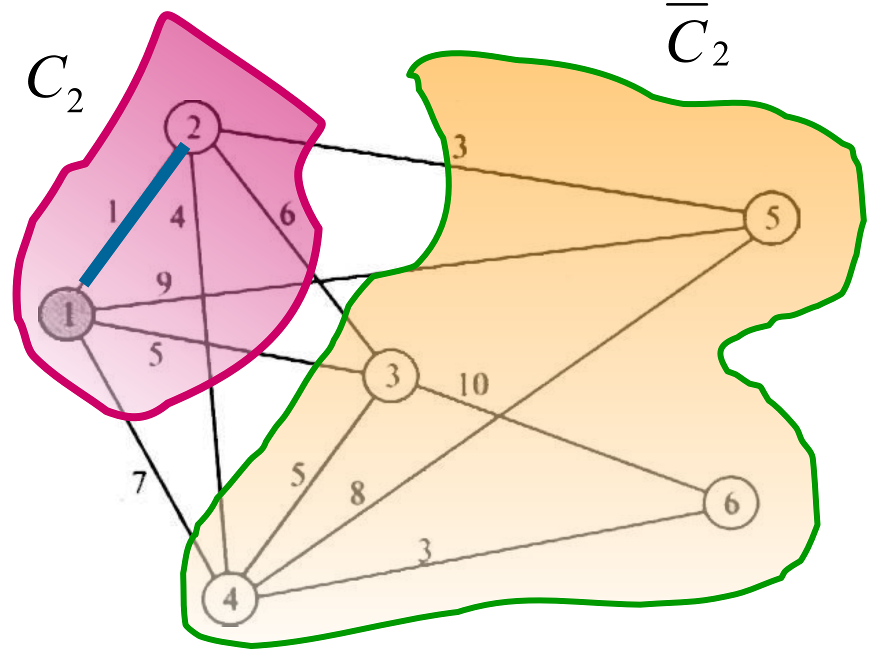
Основные этапы.

2. Примем k=2. Ищем самую короткую дугу, соединяющую узел 1 с другими узлами: min(1, 9, 5, 7) = 1. Тогда $j^*=2$

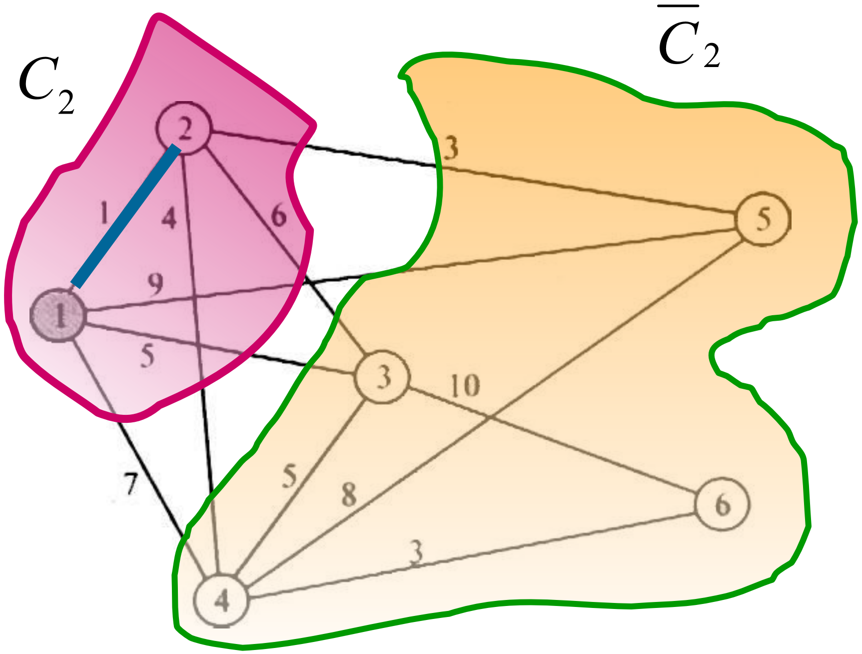
$$C_2 = \{1,2\} \quad \overline{C}_2 = \{3,4,5,6\}$$



Итерация 1



Итерация 2



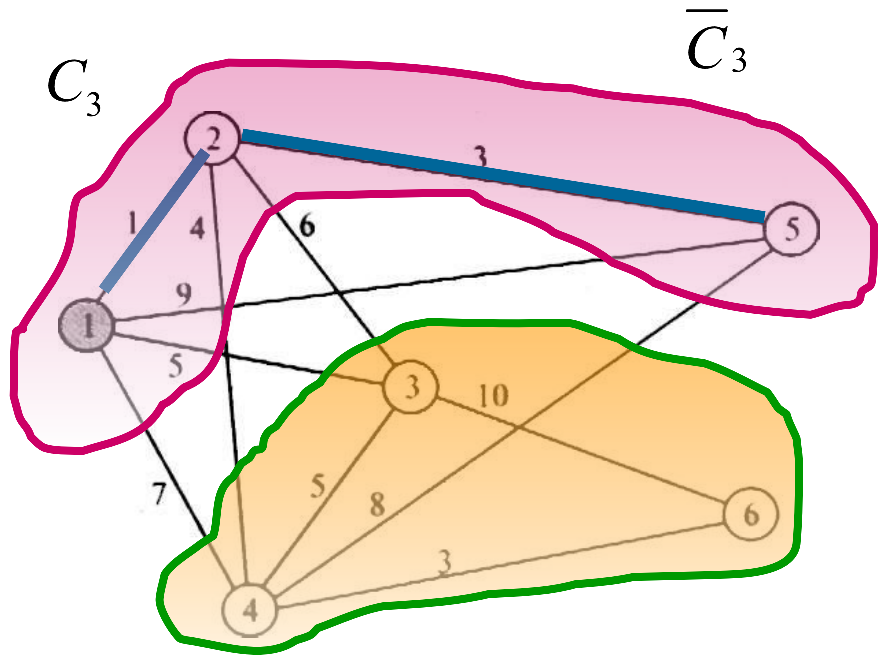
Итерация 2

Основные этапы.

3. Примем k=3. Ищем самую короткую дугу, соединяющую узел 1 или узел 2 с другими узлами:
$\min_1(9, 5, 7) = 5$, $\min_2(3, 6, 4) = 3$,
$\min(\min_1, \min_2) = \min(5, 3) = 3$.

Тогда $j^*=5$

$$C_3 = \{1,2,5\}$$
$$\overline{C}_3 = \{3,4,6\}$$



Итерация 3

Основные этапы.

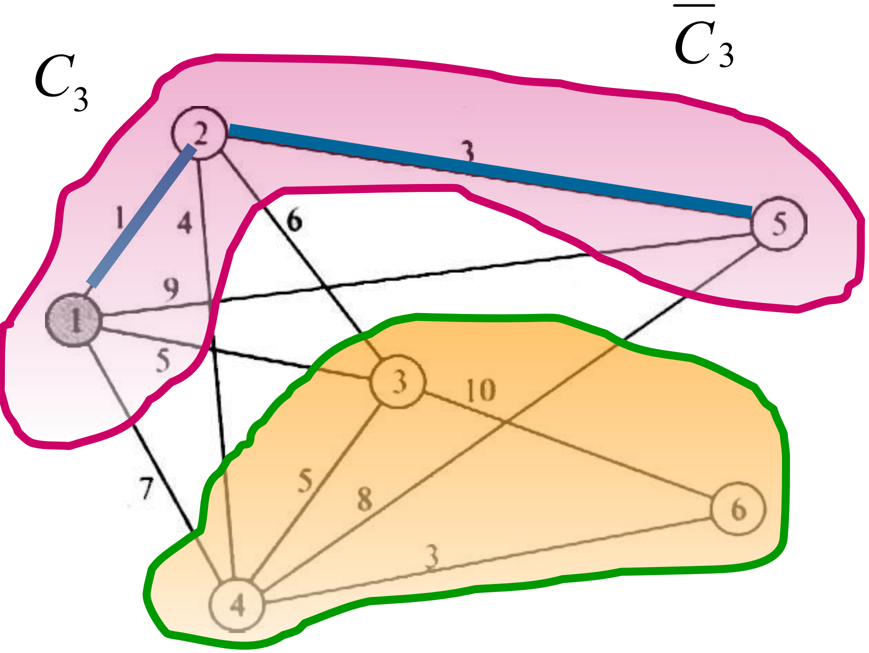
4. Примем k=4. Ищем самую короткую дугу, соединяющую узел 1, 2 или узел 5 с другими узлами:

 $\min_1(5, 7) = 5$, $\min_2(6, 4) = 4$,

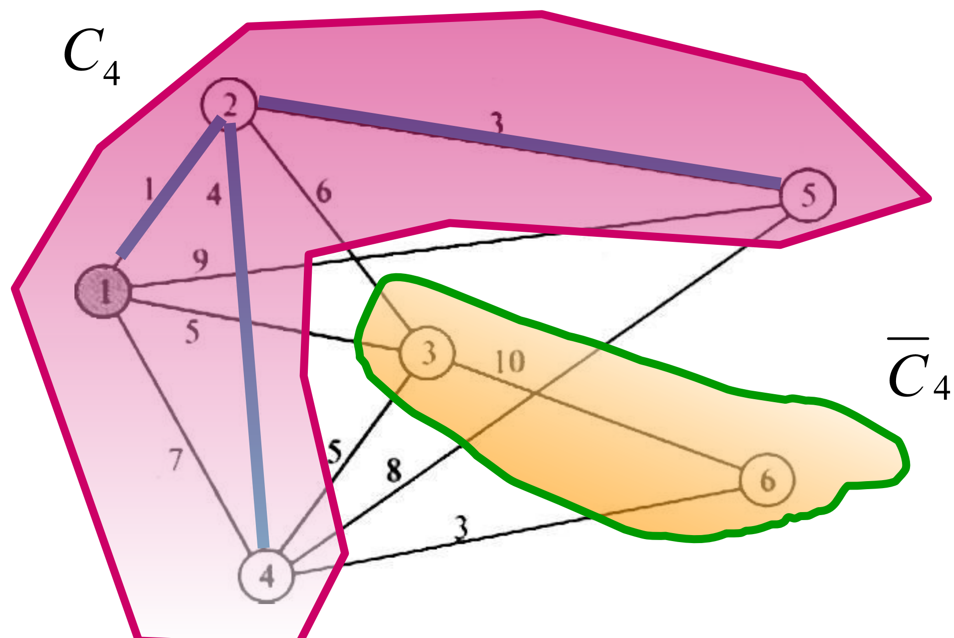
 $\min_5(8) = 8$,

 $\min(\min_1, \min_2, \min_5) = \min(5, 4{,}8) = 4$.

 Тогда j^*=4 $C_4 = \{1,2,5,4\}$ $\overline{C}_4 = \{3,6\}$



Итерация 3



Итерация 4

Основные этапы.

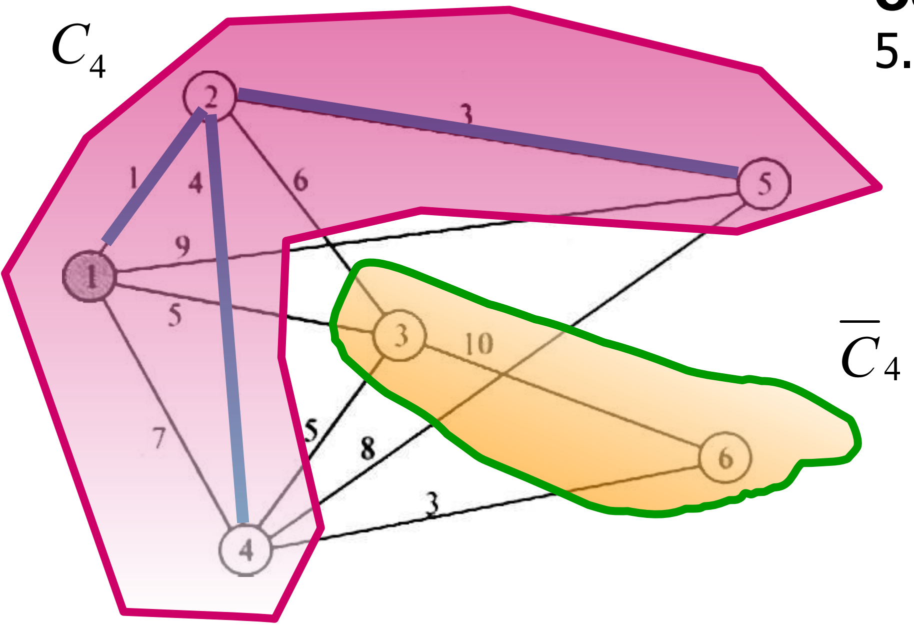
5. Примем k=5. Ищем самую короткую дугу, соединяющую узел 1, 2, 4 или узел 5 с другими узлами:

$min_1(5) = 5$, $min_2(6) = 6$, $min_4(5,3)=3$,

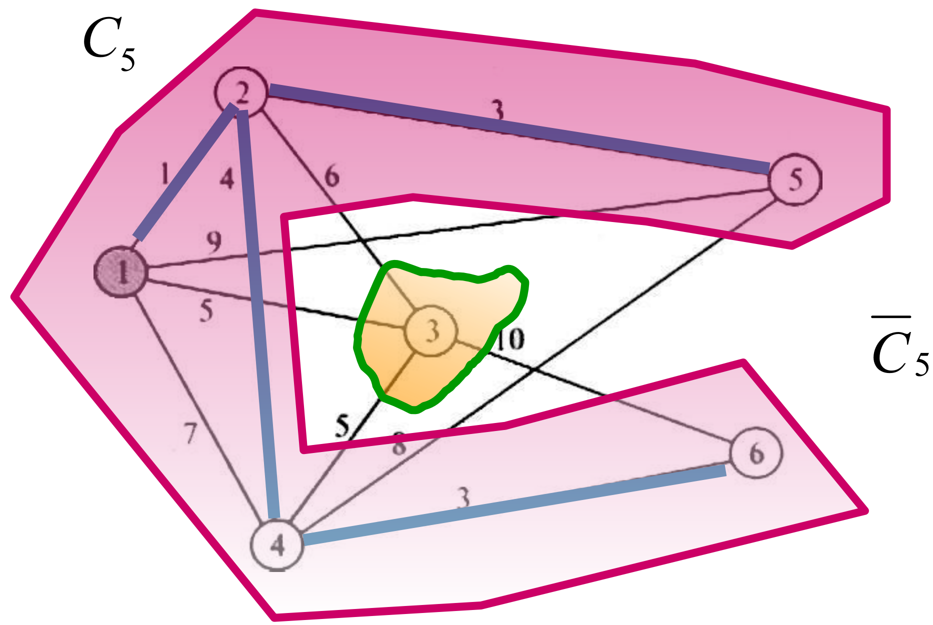
$min(min_1, min_2, min_4) = min(5,6,3) = 3.$

Тогда $j^*=6$

$$C_5 = \{1,2,5,4,6\} \quad \overline{C}_5 = \{3\}$$



Итерация 4



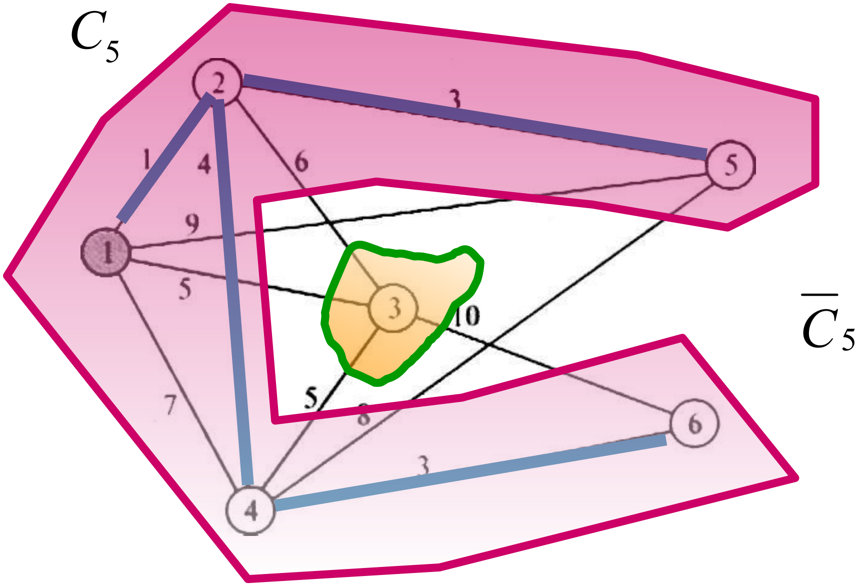
Итерация 5

Основные этапы.

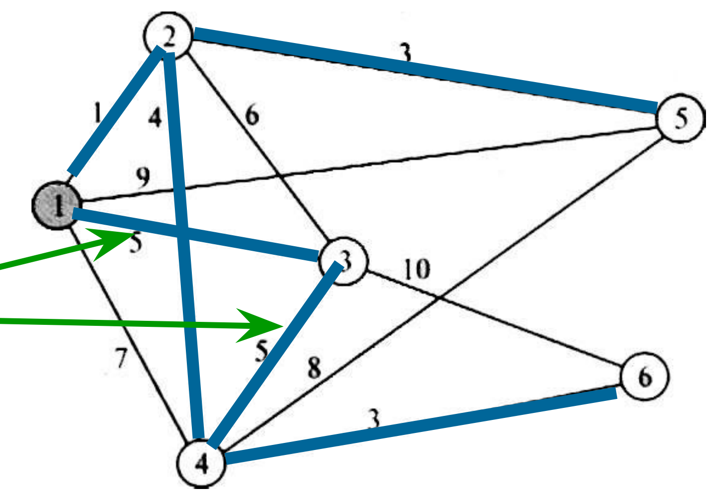
6. Примем k=6. Ищем самую короткую дугу, соединяющую узел 1, 2, 4,5 или узел 6 с другими узлами:
   $\min_1(5) = 5$, $\min_2(6) = 6$, $\min_4(5)=5$, $\min_6(10) = 10$,
   $\min(\min_1, \min_2, \min_4, \min_6) = \min(5,6,5,10) = 5.$

   Тогда $j^*=3$

$$C_6 = \{1,2,5,4,6,3\} \quad \overline{C}_6 = \varnothing$$



Итерация 5



Итерация 6

Альтернативные ребра (но реализовано может быть только **одно** - на выбор!)

Минимальная длина кабеля для построения сети **L = 1+3+4+3+5=16.**

1.1.3. Задание для самостоятельной работы

Решить задачу из Примера 1 при следующих условиях:

а) узлы 1 и 2 связаны кабелем длиной 8 км;

б) узлы 3 и 5 связаны кабелем длиной 2 км;

в) узлы 2 и 6 связаны кабелем длиной 4 км;

г) узлы 5 и 6 связаны кабелем длиной 2 км;

д) узлы 2 и 5 не связаны.

1.2 Сетевые модели: Алгоритмы поиска кратчайшего пути

Представленные здесь алгоритмы поиска кратчайшего пути справедливы как для сетей, имеющих циклы, так и для сетей, не имеющих циклов. Наиболее распространены следующие алгоритмы:

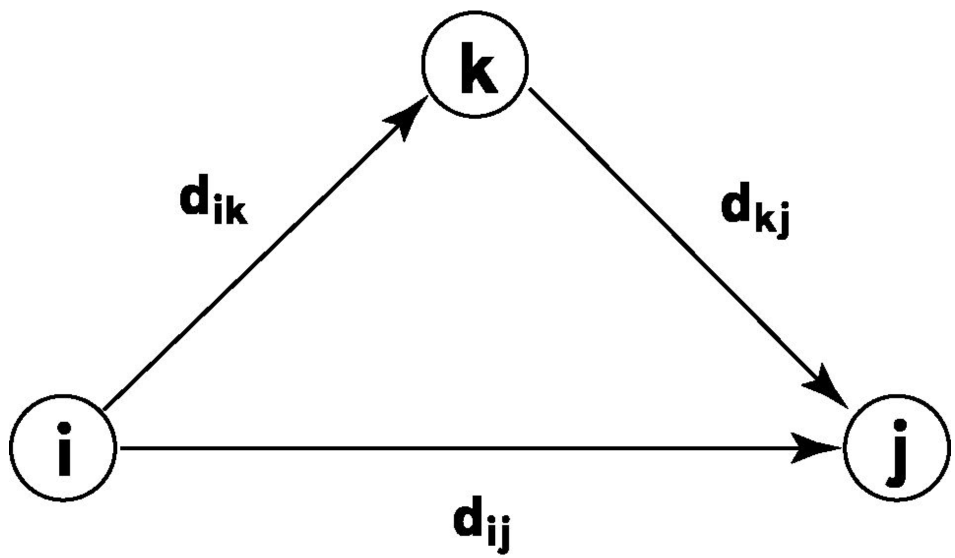
1. Алгоритм Дейкстры.

2. Алгоритм Флойда.

Алгоритм Дейкстры разработан для поиска кратчайшего пути между заданным исходным узлом и любым другим узлом сети.

Алгоритм Флойда более универсален, поскольку он в итоге дает результат, который позволяет одновременно найти минимальные пути между любыми двумя узлами сети.

В алгоритме Флойда сеть, состоящая из **n** узлов, представлена в виде квадратной матрицы с **n** строками и **n** столбцами. Элемент (*i, j*) этой матрицы равен расстоянию $\mathbf{d_{ij}}$ от узла *i* к узлу *j*, которое имеет конечное значение, если существует дуга (*i, j*), и равен бесконечности в противном случае.
Покажем сначала основную идею метода Флойда. Пусть есть три узла *i, j ,k* и заданы расстояния между ними



Если выполняется неравенство $\mathbf{d_{ik} + d_{kj} < d_{ij}}$, то целесообразно заменить путь $i \rightarrow j$ путем $i \rightarrow k \rightarrow j$. Такая замена (далее ее будем условно называть треугольным оператором Флойда) выполняется систематически в процессе выполнения алгоритма Флойда.

Этап 0 (инициализация). Определяем начальную матрицу расстояний $\mathbf{D_0}$ и матрицу последовательности узлов $\mathbf{S_0}$. Диагональные элементы обеих матриц помечаются знаком "—", показывающим, что эти элементы в вычислениях не участвуют. Полагаем *k=1*.

$\mathbf{D_0}$ =

	1	2	…	j	…	n
1	-	d_{12}	…	d_{1j}	…	d_{1n}
2	d_{21}	-	…	d_{2i}	…	d_{2n}
…	…	…	…	…	…	…
i	d_{i1}	d_{i2}	…	d_{ij}	…	d_{in}
…	…	…	…	…	…	…
n	d_{n1}	d_{n2}	…	d_{nj}	…	-

$\mathbf{S_0}$ =

	1	2	…	j	…	n
1	-	2	…	j	…	n
2	1	-	…	j	…	n
…	…	…	…	…	…	…
i	1	2	…	j	…	n
…	…	…	…	…	…	…
n	1	2	…	j	…	-

Основной этап k. Задаем ст
столбец. Двигаясь построчно
треугольного оператора Фло
элементов ведущей строки, в
элемента **ij** выполняется неравенство

Условия означают:
$i \neq k$ – на k-м шаге исключаем k-ю строку
$j \neq k$– на k-м шаге исключаем k-й столбец,
$i \neq j$ – не обрабатываем диагональные элементы

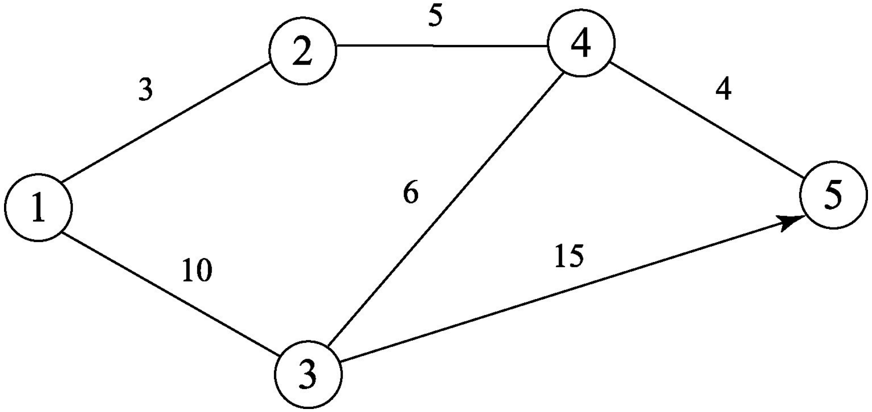
$$\mathbf{d_{ik} + d_{kj} < d_{ij},\ \ (i \neq k,\ j \neq k,\ i \neq j),}$$

то делаем следующее:
1) создаем матрицу $\mathbf{D_k}$ путем замены в матрице $\mathbf{D_{k-1}}$ элемента $\mathbf{d_{ij}}$ суммой $\mathbf{d_{ik} + d_{kj}}$;
2) создаем матрицу $\mathbf{S_k}$, меняя в матрице $\mathbf{S_{k\_1}}$ элемент $\mathbf{s_{ij}}$ на **k**.
Полагаем $k = k + 1$ и повторяем этап *k.*

В итоге осуществляется **n** аналогичных основных этапов. При этом возможно, что на k-й итерации не будет никаких изменений матриц $\mathbf{D_{k-1}}$ и $\mathbf{S_{k-1}}$.
После реализации **n** этапов алгоритма определение по матрицам $\mathbf{D_n}$ и $\mathbf{S_n}$ кратчайшего пути между узлами *i* и *j* выполняется по следующим правилам:

1. Расстояние между узлами *i* и *j* равно элементу $\mathbf{d_{ij}}$ в матрице $\mathbf{D_n}$.
2. Промежуточные узлы пути от узла *i* к узлу *j* определяем по матрице $\mathbf{S_n}$.
Пусть $\mathbf{s_{ij}}$ = k, тогда имеем путь $i \rightarrow k \rightarrow j$. Если далее $\mathbf{s_{ik}}$ = k и $\mathbf{s_{kj}}$= 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*.

Пример 17. Найдем для сети, показанной ниже, кратчайшие пути между любыми двумя узлами. Расстояния 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 допускают движение в обе стороны.



Решение: согласно Этапу 0 имеем

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	-

Полагаем k=1 и переходим к Основному этапу 1.

Основной этап 1

$D_0 =$

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

Проверяем выполнение неравенства
$\mathbf{d_{ik} + d_{kj} < d_{ij}}$,
для $k=1$, $i \neq 1$, $j \neq 1$, $i \neq j$.

$i=2$, $j = 3$: $d_{21} + d_{13} = 3+10=\mathbf{13<d_{23}}=\infty$ - выполняется
$i=2$, $j = 4$: $d_{21} + d_{14} = 3+\infty < d_{24}=5$ - не выполняется
$i=2$, $j = 5$: $d_{21} + d_{15} = 3+\infty < d_{25}=\infty$ - не выполняется

⇒ $\mathbf{d_{23} = 13}$, $\mathbf{s_{23} = 1}$

$i=3$, $j = 2$: $d_{31} + d_{12} = 10+3=\mathbf{13<d_{32}}=\infty$ - выполняется
$i=3$, $j = 4$: $d_{31} + d_{14} = 10+\infty < d_{34}=6$ - не выполняется
$i=3$, $j = 5$: $d_{31} + d_{15} = 10+\infty < d_{35}=15$ - не выполняется

⇒ $\mathbf{d_{32} = 13}$, $\mathbf{s_{32} = 1}$

$i=4$, $j = 2$: $d_{41} + d_{12} = \infty +3 < d_{42}=5$ - не выполняется
$i=4$, $j = 3$: $d_{41} + d_{13} = \infty + 10 < d_{43}=6$ - не выполняется
$i=4$, $j = 5$: $d_{41} + d_{15} = \infty + \infty < d_{45}=4$ - не выполняется

$i=5$, $j = 2$: $d_{51} + d_{12} = \infty +3 < d_{52} = \infty$ - не выполняется
$i=5$, $j = 3$: $d_{51} + d_{13} = \infty + 10 < d_{53}= \infty$ - не выполняется
$i=5$, $j = 4$: $d_{51} + d_{14} = \infty + \infty < d_{54}=4$ - не выполняется

Соответственно создаем матрицы **D_1** и **S_1**.

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	-

Полагаем k=2 и переходим к Основному этапу 2.

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	-

Проверяем выполнение неравенства

$\mathbf{d_{ik} + d_{kj} < d_{ij}}$,

для k=2, i≠2, j ≠ 2, i ≠ j.

i=1, j = 3: $d_{12} + d_{23}$ = 3+13= 16 < d_{13}=10 - не выполняется

i=1, j = 4: $d_{12} + d_{24}$ = 3+ 5 = **8 < d_{14}** = ∞ - выполняется

i=1, j = 5: $d_{12} + d_{25}$ = 3+ ∞ < d_{15} = ∞ - не выполняется

i=3, j = 1: $d_{32} + d_{21}$ = 13+3= 16 < d_{31} =10 - не выполняется

i=3, j = 4: $d_{32} + d_{24}$ = 13+5 = 18 < d_{34} = 6 - не выполняется

i=3, j = 5: $d_{32} + d_{25}$ = 13+ ∞ < d_{35} = 15 - не выполняется

i=4, j = 1: $d_{42} + d_{21}$ = 5+3 = **8 < d_{41}** = ∞ выполняется

i=4, j = 3: $d_{42} + d_{23}$ = 5+13 = 18 < d_{43} = 6 - не выполняется

i=4, j = 5: $d_{42} + d_{25}$ = 5+ ∞ < d_{45} = 4 - не выполняется

i=5, j = 1: $d_{52} + d_{21}$ = ∞ + 3 < d_{51} = ∞ - не выполняется

i=5, j = 3: $d_{52} + d_{23}$ = ∞ + 13 < d_{53} = ∞ - не выполняется

i=5, j = 4: $d_{52} + d_{24}$ = ∞ + 5 < d_{54} = 4 - не выполняется

d_{14} = 8, s_{14} = 2

d_{41} = 8, s_{41} = 2

Соответственно создаем 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	-

Полагаем k=3 и переходим к Основному этапу 3.

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	-

Проверяем выполнение неравенства

$\mathbf{d_{ik} + d_{kj} < d_{ij}}$,

для k=3, i≠3, j ≠ 3, i ≠ j.

i=1, j = 2: $d_{13} + d_{32}$ = 10+13= 23 < d_{12}=3 - не выполняется

i=1, j = 4: $d_{13} + d_{34}$ = 10+ 6 = 16 < d_{14} =8 - не выполняется

i=1, j = 5: $d_{13} + d_{35}$ = 10+15= **25 <** $\mathbf{d_{15}}$ =∞ - выполняется

d_{15} = 25, s_{15} = 3

i=2, j = 1: $d_{23} + d_{31}$ = 13+10= 23 < d_{21} =3 - не выполняется

i=2, j = 4: $d_{23} + d_{34}$ = 13+6 = 19 < d_{24} =5 - не выполняется

i=2, j = 5: $d_{23} + d_{35}$ = 13+15 = **28 <** $\mathbf{d_{25}}$ = ∞ - выполняется

d_{25} = 28, s_{25} = 3

i=4, j = 1: $d_{43} + d_{31}$ = 6+10 = 16 < d_{41} =8 - не выполняется

i=4, j = 2: $d_{43} + d_{32}$ = 6+13 = 19 < d_{42} = 5 - не выполняется

i=4, j = 5: $d_{43} + d_{35}$ = 6+15 = 21 < d_{45} = 4 - не выполняется

i=5, j = 1: $d_{53} + d_{31}$ = ∞+10 < d_{51} = ∞ - не выполняется

i=5, j = 2: $d_{53} + d_{32}$ = ∞+13 < d_{52} = ∞ - не выполняется

i=5, j = 4: $d_{53} + d_{34}$ = ∞+ 6 < d_{54} =4 - не выполняется

Соответственно создаем 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	-

Полагаем k=4 и переходим к Основному этапу 4.

$D_3 =$

	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	-

Проверяем выполнение неравенства
$\mathbf{d_{ik} + d_{kj} < d_{ij}}$,
для k=4, i≠4, j ≠ 4, i ≠ j.

i=1, j = 2: $d_{14} + d_{42} = 8+5 = 13 < d_{12} = 3$ - не выполняется

i=1, j = 3: $d_{14} + d_{43} = 8+6 = 14 < d_{13} = 10$ - не выполняется

i=1, j = 5: $d_{14} + d_{45} = 8+4 = \mathbf{12 < d_{15}} = 25$ - выполняется

d_{15} = 12, s_{15} = 4

i=2, j = 1: $d_{24} + d_{41} = 5+8 = 13 < d_{21} = 3$ - не выполняется

i=2, j = 3: $d_{24} + d_{43} = 5+6 = \mathbf{11 < d_{23}} = 13$ - выполняется

d_{23} = 11, s_{23} = 4

i=2, j = 5: $d_{24} + d_{45} = 5+4 = \mathbf{9 < d_{25}} = 28$ - выполняется

d_{25} = 9, s_{25} = 4

i=3, j = 1: $d_{34} + d_{41} = 6+8 = 14 < d_{31} = 10$ - не выполняется

i=3, j = 2: $d_{34} + d_{42} = 6+5 = \mathbf{11 < d_{32}} = 13$ - выполняется

d_{32} = 11, s_{32} = 4

i=3, j = 5: $d_{34} + d_{45} = 6+4 = \mathbf{10 < d_{35}} = 15$ - выполняется

d_{35} = 10, s_{35} = 4

i=5, j = 1: $d_{54} + d_{41} = 4+8 = \mathbf{12 < d_{51}} = \infty$ - выполняется

d_{51} = 12, s_{51} = 4

i=5, j = 2: $d_{54} + d_{42} = 4+5 = \mathbf{9 < d_{52}} = \infty$ - выполняется

d_{52} = 9, s_{52} = 4

i=5, j = 3: $d_{54} + d_{43} = 4+6 = \mathbf{10 < d_{53}} = \infty$ - выполняется

d_{53} = 10, s_{53} = 4

Соответственно создаем матрицы D_4 и S_4

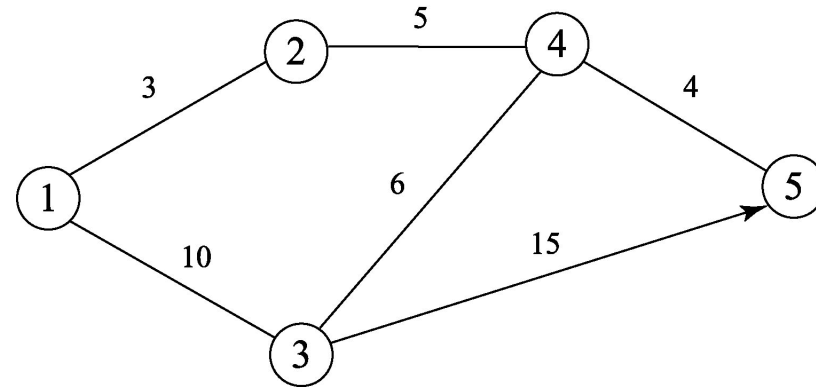
D_4 =

	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	-

Полагаем k=5 и переходим к Основному этапу 5. На этом этапе нет изменений в матрицах: $\mathbf{D_5} = \mathbf{D_4}$, $\mathbf{S_5} = \mathbf{S_4}$. Вычисления закончены.



Для определения соответствующих маршрутов напомним, что сегмент маршрута состоит из ребра (i, j) только тогда, когда $\mathbf{s_{ij}}$=j. В противном случае узлы i и j связаны, по крайней мере, через один промежуточный узел.
Например, поскольку $\mathbf{s_{15}} = 4$ и $\mathbf{s_{45}} = 5$, сначала кратчайший маршрут между узлами 1 и 5 будет иметь вид 1 → 4 → 5. Но так как $\mathbf{s_{14}} \neq 4$, узлы 1 и 4 в определяемом кратчайшем 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 $\mathbf{s_{14}} = 2$ и $\mathbf{s_{24}} = 4$, поэтому маршрут 1 → 4 заменяем 1 → 2 → 4. Поскольку $\mathbf{s_{12}} = 2$ и $\mathbf{s_{24}} = 4$, других промежуточных узлов нет.
Комбинируя определенные сегменты маршрута, окончательно получаем следующий кратчайший путь от узла 1 до узла 5: 1 → 2 → 4 → 5. Длина этого пути равна 12.

Задача для самостоятельного решения