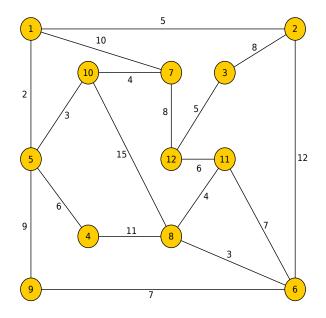


Сформируем список ребер в виде матрицы

v1	v2	weight
1	2	5
1	5	2
1	7	10
2	3	8
2	6	12
3	12	5
4	5	6
4	8	11
5	9	9
5	10	3
6	8	3
6	9	7
6	11	7
7	10	4
7	12	8
8	10	15
8	11	4
11	12	6



# Отсортируем ребра по весу

v1	v2	weight
1	5	2
5	10	3
6	8	3
7	10	4
8	11	4
1	2	5
3	12	5
4	5	6
11	12	6
6	9	7
6	11	7
2	3	8
7	12	8
5	9	9
1	7	10
4 2	8	11
	6	12
8	10	15

1		(2) Компоненты с		омпоненты связности	
				Nº	Вершины, входящие в состав
	10	7	3	1	1
				2	2
				3	3
				4	4
				5	5
5		12	11	6	6
		•		7	7
				8	8
				9	9
	4	8		10	10
				11	11
_				_ 12	12
9				6	

Создадим список компонент связности. В начале алгоритма число компонент равно числу вершин графа

 $AdComps < - \ list( \ c(1), \ c(2), \ c(3), \ \dots \ , \ c(11), \ c(12) \ )$ 

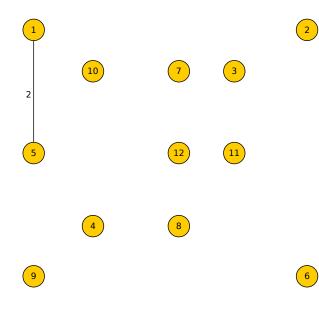
v1	v2	weight
1	5	
1 5	10	3
6	8	3
7	10	4
8	11	4
1 3	2	2 3 4 4 5 5 6 6 7 8
	12	5
4	5	6
11	12	6
6	9	7
6	11	7
2	3	8
7 5	12	8
5	9	9
1	7	10
1 4 2 8	8	11
2	6	12
8	10	15

Количество компонент связности:

12

Алгоритм завершится, когда либо

- а) количество компонент связности станет равно 1;
- **б)** закончатся ребра (в случае несвязного графа G).



Nº	Вершины, входящие в состав
1	{ <u>1, 5</u> }
2	2
3	3
4	4
6	6
7	7
8	8
9	9
10	10
11	11
12	12

Вершины 1 и 5 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 5. Удаляем компоненту связности 5.

AdComps < - list( c( 1, AdComps[[5]] ), c(2), ..., c(12) ) AdComps < - AdComps[ -5 ]

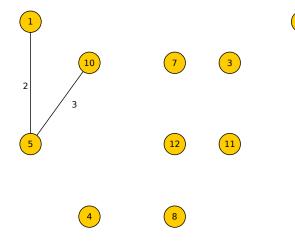
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	<b>2</b> ←
5	10	3
6	8	3
7	10	4
8	11	4
1	2	<b>2</b> ← 3 3 4 4 5 5
1 3	12	5
4	5	6
11	12	6
6	9	7
6	11	7
2	3	8
7	12	8
5	9	9
1	7	10
4	8	11
1 4 2 8	6	12
8	10	15

# Удаленные компоненты связности

№ Вершины, входящие в состав

Новое количество компонент связности:



Nº	Вершины, входящие в состав
1	{1, 5,10}
2	2
3	3
4	4
5	6
6	7
7	8
8	9
10	11
11	12

Вершины 5 и 10 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 9. Удаляем компоненту связности 9.

AdComps < - list( c( 1, 5, AdComps[[ 9 ]] ), c(2),...,c(12) ) AdComps < - AdComps[ -9 ]

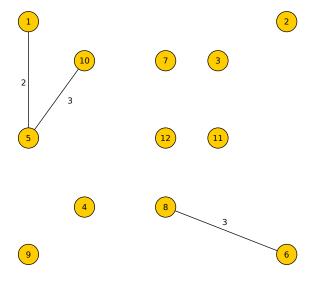
# Отсортированный по весу список ребер графа

v1	v2	weight
1	5	2
5	10	3 ←
6	8	3
7	10	4
8	11	4
1	2	5
3	12	2 3 4 4 5 5
4	5	6
11	12	6 6 7 7 8 8
6	9	7
6	11	7
2	3	8
7	12	8
5	9	9
1	7	10
4	8	11
2	6	12
8	10	15

## Удаленные компоненты связности

№ Вершины, входящие в состав 5 5 **9 10** 

Новое количество компонент связности:



Вершины, входящие в состав	Nº
{1, 5,10}	1
2	2
3	3
4	4
{6, 8 }	5
7	6
9	8
11	9
12	10

Вершины 6 и 8 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 5 и 7. Удаляем компоненту связности 7.

AdComps < - list( c(1, 5, 10),...,c(6, AdComps[[ 7 ]]), ..., c(12) ) AdComps < - AdComps[ -7 ]

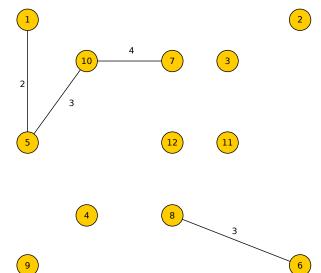
Отсортированный по весу список ребер графа

v1	v2	weight
1	5	2
5	10	3
6	8	3 ←
7	10	2 3 3 4 4 5
8	11	4
1	2	
3	12	5
4	5	6
11	12	6
6	9	7
6	11	7
2	3	8
7 5	12	8
5	9	9
1	7	10
4	8	11
2	6	12
8	10	15

#### Удаленные компоненты связности

№ Вершины,	входящие в состав
5	5
9	10
7	8

Новое количество компонент связности:



№ Вершины, в	ходящие в состав
1	{1, 5, 10, 7}
2	2
3	3
4	4
5	{6, 8}
7	9
8	11
9	12

Вершины 10 и 7 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 6. Удаляем компоненту связности 6.

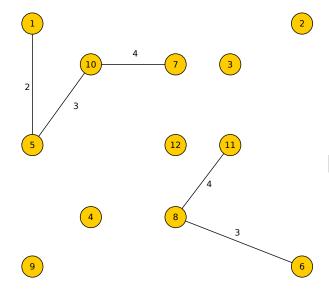
AdComps < - list( c(1, 5, 10, AdComps[[ 6 ]]), c(2),..., c(12) ) AdComps < - AdComps[-6]

## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	2
5	10	2 3 3
6	8	
7	10	<b>4</b> ←
8	11	<b>4</b> ← 4 5
1	2	
3	12	5
4	5	6
11	12	6
6	9	7
6	11	7
2	3	8
7	12	8
5	9	9
1	7	10
4	8	11
2	6	12
8	10	15

Новое количество компонент связности: 8

Nº	Вершины, входящие в состав
5	5
9	10
7	8
6	7



№ Вершины, входящие в состав	3
1 {1, 5, 10, 7	}
2	2
3	3
4	4
5 <b>{6, 8, 11</b>	}
6	9
8 1	2

Вершины 8 и 11 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 5 и 7. Удаляем компоненту связности 7.

 $\label{eq:adComps} $$AdComps < -list( c(1, 5, 10, 7), ..., c(6, 8, AdComps[[ 7 ]]), ..., c(12) ) $$AdComps < -AdComps[ -7 ] $$$ 

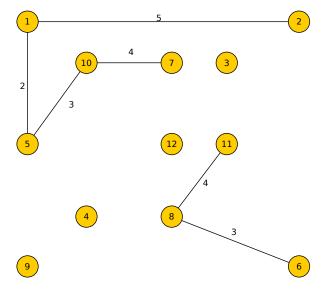
## Отсортированный по весу список ребер графа

v1	v2	weight	
1	5	2	
5	10	2 3 3	
6	8	3	
7	10	4	
8	11	4	<u>_</u>
1	2	4 <b>4</b> 5 5	
3	12	5	
4	5	6	
11	12	6	
6	9	7	
6	11	7	
2	3	8	
7	12	8	
5	9	9	
1	7	10	
4	8	11	
2	6	12	
8	10	15	

Новое количество компонент связности:

7

 N.	Dan
Nº	Вершины, входящие в состав
5	5
9	10
7	8
6	7
7	11



Вершины 1 и 2 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 2. Удаляем компоненту связности 2.

AdComps < - list( c(1, 5, 10, 7, AdComps[[ 2 ]]),..., c(12) ) AdComps < - AdComps[ -2 ]

## Отсортированный по весу список ребер графа

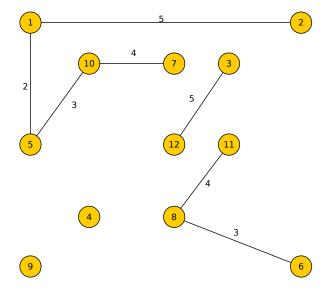
v1		weight	
1	. 5	2	
1	10	2 3 3	
6	8	3	
7	10	4	
8	11	4	
1		4 <b>5</b> 5	
3	12	5	
4		6	
11	. 12	6	
6	9	7	
6	11	7	
2	3	8	
7	12	8	
5	9	9	
1		10	
4	8	11	
2		12	
8	10	15	

### Удаленные компоненты связности

Nº	Вершины, входящие в состав
5	5
9	10
7	8
6	7
7	11
2	2

Новое количество компонент связности:

b



Nº	Вершины, входящие в состав
1	{1, 5,10, 7, 2}
2	{ 3, 12 }
3	4
4	{6, 8, 11 }
5	9

Вершины 3 и 12 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 2 и 6. Удаляем компоненту связности 6.

AdComps < - list( c(1, 5, 10, 7, 2), c(3, AdComps[[ 6 ]]),...,c(12) ) AdComps < - AdComps[ -6 ]

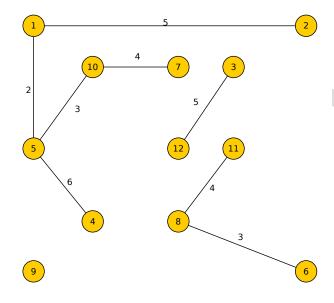
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	2
1 5 6	10	3
	8	3
7	10	4
8	11	4
1 <b>3</b>	2	5
	12	weight 2 3 3 4 4 5 5 6
4	5	6
11	12	6
6	9	7
6	11	7
6 2 7	3	8
7	12	8
5	9	9
1	7	10
4 2	8	11
2	6	12
8	10	15

Новое количество компонент связности:

5

- Homes in the manufacture in a particular in the partin the particular in the particular in the particular in the parti		
Nº	Вершины, входящие в состав	
5	5	
9	10	
7	8	
6	7	
7	11	
2	2	
6	12	



Nº	Вершины, входящие в состав
1	{1, <mark>5</mark> ,10, 7, 2, 4 }
2	{ 3, 12 }
4	{6, 8, 11 }
5	9

Вершины 4 и 5 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 3. Удаляем компоненту связности 3.

AdComps < - list( c(1, 5, 10, 7, 2, AdComps[[ 3 ]]), c(3, 12),..., c(9) ) AdComps < - AdComps[ -3 ]

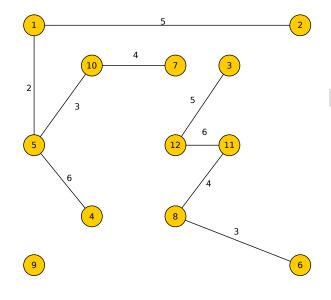
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	
1 5	10	3
6	8	2 3 3 4
7	10	4
8	11	4
1	2	4 5 5
3	12	5
4	5	<b>6</b> ← 6
11	12	6
6	9	7
6	11	7
2	3	8
7	12	8
5	9	9
1	7	10
4	8	11
2	6	12
8	10	15

#### Удаленные компоненты связности

	- 1
Вершины, входящие в состав	Nº
5	5
10	9
8	7
7	6
11	7
2	2
12	6
4	3

Новое количество компонент связности:



Nº	Вершины, входящие в состав
1	{1, 5, 10, 7, 2, 4}
2	{ 3, <mark>12</mark> , 6, 8, 11 }
4	9

Вершины 11 и 12 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 2 и 3. Удаляем компоненту связности 3.

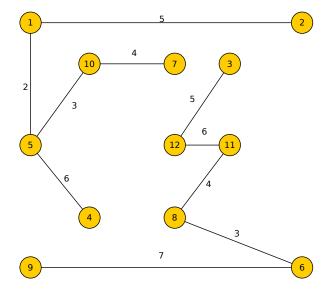
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	2
1 5 6	10	3
6	8	3
7	10	4
8	11	4
1	2	2 3 3 4 4 5 5 6 <b>6</b> 7
1 3 4	12	5
	5	6
11	12	6 ←
6	9	
6	11	7
6 2 7 5	3	8
7	12	8
5	9	9
1	7	10
4	8	11
1 4 2 8	6	12
8	10	15

# Удаленные компоненты связности

Nº	Вершины, входящие в состав
5	5
9	10
7	8
6	7
7	11
2	2
6	12
3	4
3	{6, 8, <mark>11</mark> }

Новое количество компонент связности:



Вершины 6 и 9 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 2 и 3. Удаляем компоненту связности 3.

AdComps < - list( c(1,..., 4), c(3,...,11, AdComps[[ 3 ]]), c(9) ) AdComps < - AdComps[ -3 ]

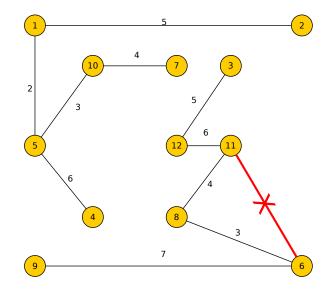
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	
5	10	2 3
6	8	3
7	10	4
8	11	4
1 3	2	5
3	12	5
4	5	6
11	12	6
6	9	<b>7</b> ←
6	11	<b>7</b> ← 7
2	3	8
6 2 7 5	12	8
5	9	9
1	7	10
1 4 2	8	11
2	6	12
8	10	15

#### Удаленные компоненты связности

	- 1 1
Вершины, входящие в состав	Nº
5	5
10	9
8	7
7	6
11	7
2	2
12	6
4	3
{6, 8, 11}	3
9	3

Новое количество компонент связности:



Вершины 6 и 11 входят в одну компоненту связности!!! Ребро (6, 11) НЕ добавляется в дерево

 $\mbox{AdComps} < - \mbox{ list( } \mbox{c(1, 5, 10, 7, 2, 4), c(3, 12, 6, 8, 11, 9) ) }$ 

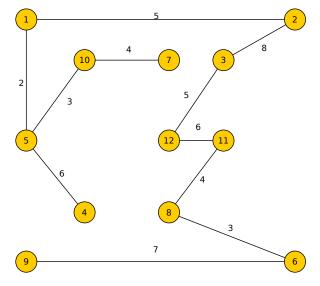
## Отсортированный по весу список ребер графа

v1	v2	weight
1	5	
5	10	2 3 3
6	8	3
7	10	4
8	11	4
1	2	5
3	12	5
4	5	6
11	12	6
6	9	7
6	11	4 4 5 5 6 6 7 <b>7</b> 8
2 7	3	
	12	8
5	9	9
1	7	10
4	8	11
2	6	12
8	10	15

Новое количество компонент связности:

2

Nº	Вершины,	входящие	в состав
5			5
9			10
7			8
6			7
7			11
2			2
6			12
3			4
3			{6, 8, 11 }
3			9



№ Вершины, входящие в состав 1 **{1, 5,10, 7, 2, 4, 3, 12, 6,8,11, 9 }** 

Вершины 2 и 3 находятся в разных компонентах связности. Добавляем ребро в дерево. Объединяем компоненты связности 1 и 2. Удаляем компоненту связности 2.

AdComps < - list( c(1,...,4, AdComps[[ 2 ]]), c(3, 12, 6,8,11, 9) ) AdComps < - AdComps[ -2 ]

## Отсортированный по весу список ребер графа

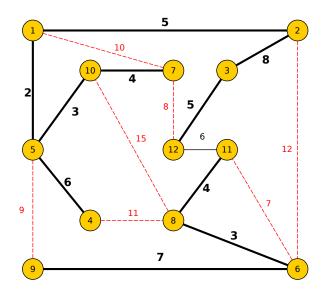
v1	v2	weight
1 5	5	
5	10	2 3 3
6	8	
7	10	4
8	11	4
1	2	5
1 3	12	5
4	5	6
11	12	6
6	9	7
6	11	7
<b>2</b> 7 5	3	<b>8</b> ← 8
7	12	8
5	9	9
1	7	10
4 2	8	11
2	6	12
8	10	15

## Удаленные компоненты связности

Nº	Вершины, входящие в состав
5	5
9	10
7	8
6	7
7	11
2	2
6	12
3	4
3	{6, 8, 11}
3	9
2	{ <mark>3</mark> , 12, 6, 8, 11, 9 }

Новое количество компонент связности:

Минимальное остовное дерево построено.



№ Вершины, входящие в состав 1 **{1, 5,10, 7, 2, 4, 3, 12, 6,8,11, 9**}

Алгоритм завершен. Красным пунктиром показаны ребра, не вошедшие в дерево

## AdComps < - list( c(1, 5, 10, 7, 2, 4, 3, 12, 6,8,11, 9) ) length ( AdComps ) == 1

Дерево построено, если количество компонент связности равно единице

	weight	v2	v1
	2	v2 5	1
	2 3 3	10	1 5
	3	8	6
	4	10	7
	4	11	8
	5	2	1
	5	12	3 4
	6	5	4
	6	12	11
	7	9	6
	7	11	6
←	7 8 8	3	2
	8	12	7
	9	9	5
	10	7	1
	11	8	1 4
	12	6	2
	15	10	8