Теория конечных графов

Алгоритм Краскала

Лектор: к.ф.-м.н., доцент кафедры
прикладной информатики и теории вероятностей РУДН
Маркова Екатерина Викторовна
markova_ev@pfur.ru

Литература

1. **Зарипова Э.Р., Кокотчикова М.Г. Лекции по дискретной математике: Теория графов. Учебное пособие. М., изд-во: РУДН, 2013, 162 с.**
2. Харари Ф. «Теория графов», М.: КомКнига, 2006. – 296 с.
3. Судоплатов С.В., Овчинникова Е.В. «Элементы дискретной математики». Учебник. М.: Инфра-М; Новосибирск: НГТУ, 2003. – 280 с.
4. Шапорев С.Д. «Дискретная математика. Курс лекций и практических занятий». СПб.: БХВ-Петербург, 2007. – 400 с.: ил.
5. Сайт кафедры прикладной информатики и теории вероятностей РУДН (информационный ресурс). Режим доступа: http://api.sci.pfu.edu.ru/ – свободный.
6. Учебный портал кафедры прикладной информатики и теории вероятностей РУДН (информационный ресурс) Режим доступа: http://stud.sci.pfu.edu.ru – для зарегистрированных пользователей.
7. **Учебный портал РУДН, раздел «Теория конечных графов» http://web-local.rudn.ru/web-local/prep/rj/index.php?id=209&p=26342**

Покрывающее дерево

Дерево T покрывает граф G, если все вершины графа G принадлежат дереву T. Такое дерево называется **покрывающим**.

Покрывающее дерево существует только для связного графа (лес для несвязного графа).

Алгоритм Краскала (1957 г.) позволяет построить такое покрывающее дерево. Алгоритм Краскала можно представить как процесс окрашивания ребер. Ребра окрашиваются в два цвета: синий и оранжевый.

Задача о распространении слухов в деревне

<u>Задача.</u> В небольшой деревушке некоторые из жителей имеют каждодневные встречи друг с другом. Может ли в этой деревне распространиться какой-либо слух?

Чтобы ответить на этот вопрос, поставим в соответствие каждому жителю деревни вершину графа. Соединим две вершины ребром, если соответствующие жители ежедневно общаются друг с другом и рассказывают все новости. При условии связности полученного таким образом графа на поставленный в задаче вопрос можно ответить положительно.

Раскраска ребер графа

Цвет

Синим цветом окрашиваются ребра, включаемые в покрывающее дерево

Оранжевым цветом окрашиваются ребра, не включаемые в покрывающее дерево, т.к. они образуют цикл с синими ребрами или являются петлями.

Упорядочивание ребер графа

1) Пронумеруем все ребра графа следующим образом:

$$e_1, e_2, ..., e_m \; ; \; |\mathbf{E}| = m \, ,$$

2) или образуем упорядоченное множество ребер $\mathbf{E} = \{(V_i, V_j) \mid \; i \le j, \;\; i, j = \overline{1, |\mathbf{V}|}\}$ для случая, когда ребра представлены через вершины. Упорядочим ребра, используя лексикографический порядок, где сначала упорядочиваются ребра по первой вершине, а потом по второй вершине. Например, $(V_1, V_2), ..., (V_1, V_5), ..., (V_2, V_5), ...$, то есть (V_i, V_j) , где $i \le j$, $i, j = \overline{1, |\mathbf{V}|}$. Например, сначала идет ребро (V_i, V_5) , потом ребро (V_i, V_6) .

Букет

Букет – множество вершин, принадлежащих одной компоненте связности.

Ребро образует цикл с ребрами, уже включенными в дерево, если обе его концевые вершины принадлежат одному букету.

Существует 2 случая связности графа:

1° Если $G = (\mathbf{V}, \mathbf{E})$ – связный граф, где $|\mathbf{V}| = n$, то построение покрывающего дерева по алгоритму Краскала заканчивается в том случае, когда количество ребер, окрашенных в синий цвет, становится равным $n-1$.

2° Если $G = (\mathbf{V}, \mathbf{E})$ – несвязный граф, где $|\mathbf{V}| = n$, то построение покрывающего дерева по алгоритму Краскала заканчивается после раскраски всех ребер графа. Число покрывающих деревьев в таком лесу будет равно числу букетов.

Построение покрывающего дерева по алгоритму Краскала

Начало: Все рёбра графа $G=(\mathbf{V},\mathbf{E})$ не окрашены и ни один из букетов не формирован.

Шаг 1. Все петли окрасить в оранжевый цвет.

Шаг 2. Из упорядоченного множества **E** выбирается первое ребро, не являющееся петлей. Это ребро окрашивается в синий цвет и формируется букет, в который включаются концевые вершины выбранного ребра.

Шаг 3. Из оставшихся ребер выбирается первое неокрашенное ребро. Если в графе такого ребра нет, следует закончить процедуру и перейти к шагу 4.

После выбора ребра возможны 4 случая:

А. Обе концевые вершины выбранного ребра принадлежат одному и тому же букету. В этом случае ребро окрашивается в оранжевый цвет.

Построение покрывающего дерева по алгоритму Краскала

Б. Одна из концевых вершин ребра принадлежит существующему букету, а другая – не принадлежит ни одному из уже сформированных букетов. В этом случае ребро окрашивается в синий цвет, и его вторая концевая вершина включается в букет, которому принадлежит первая концевая вершина.

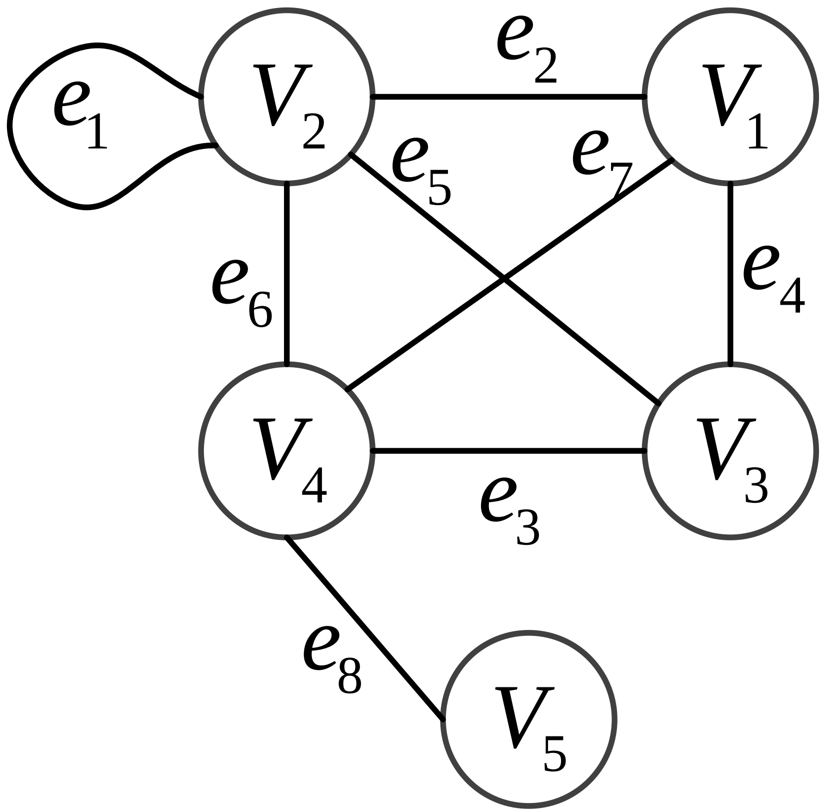
В. Концевые вершины выбранного ребра принадлежат различным букетам. В этом случае ребро окрашивается в синий цвет, а оба букета, которым принадлежат его концевые вершины, объединяются в новый букет с меньшей нумерацией.

Г. Ни одна из концевых вершин не принадлежит ни одному из сформированных букетов. В этом случае ребро окрашивается в синий цвет и формируется новый букет из концевых вершин этого ребра.

Шаг 4. Если все ребра окрашены, следует закончить алгоритм. Синие ребра образуют покрывающее дерево (лес). В противном случае вернуться к началу шага 3.

Конец алгоритма.

Пример построения покрывающего дерева по алгоритму Краскала по нумерации

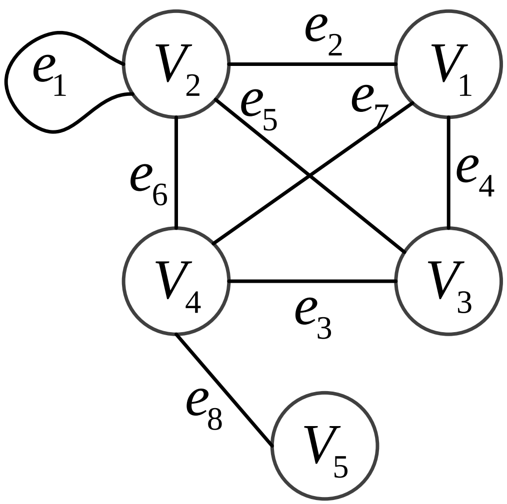


Пример 1. Построить покрывающее дерево по алгоритму Краскала для графа по нумерации

Решение для примера 1

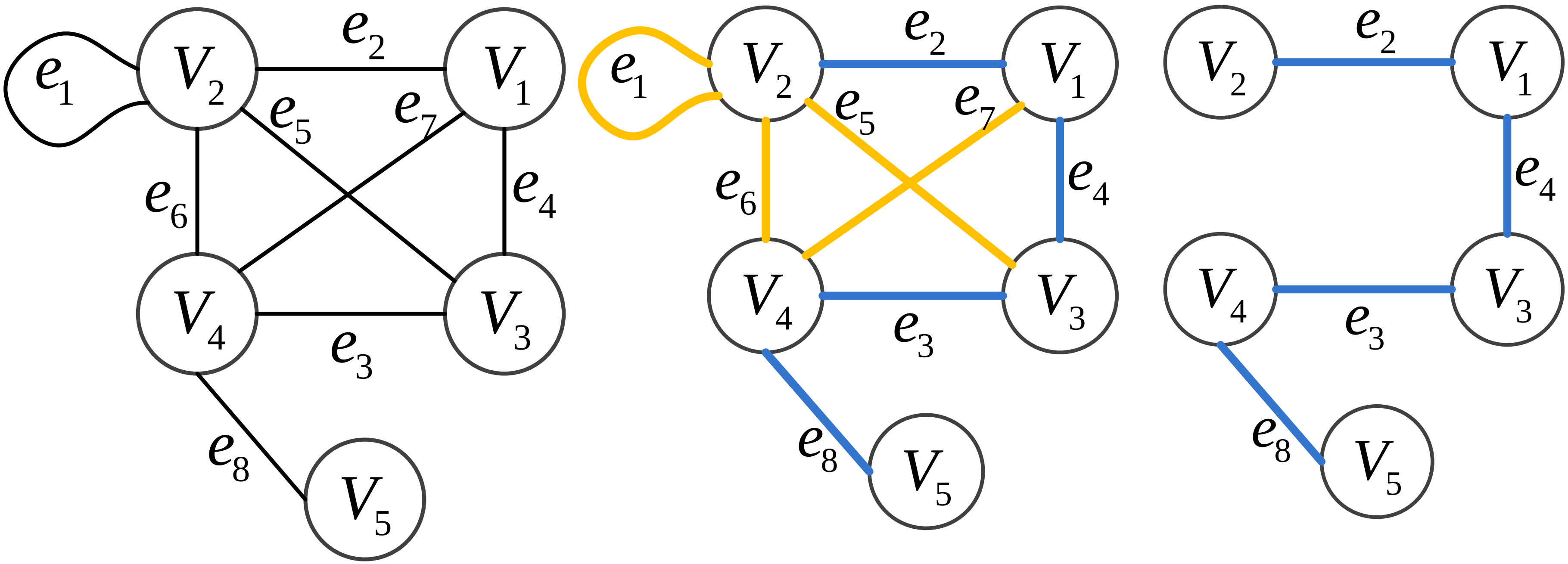
Упорядочим множество ребер: $\mathbf{E} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$.

Для поиска покрывающего дерева по алгоритму Краскала пошагово распишем процесс окраски ребер графа в виде таблицы.



Ребро	Цвет	Букет №1	Букет №2
$\varnothing$	–	Пуст	Пуст
e_1	Оранжевый	Пуст	Пуст
e_2	Синий (1)	$\{V_1, V_2\}$	Пуст
e_3	Синий (2)	$\{V_1, V_2\}$	$\{V_3, V_4\}$
e_4	Синий (3)	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_5	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_6	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_7	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_8	Синий (4)	$\{V_1, V_2, V_3, V_4, V_5\}$	Пуст

Ответ для примера 1



$$T = \{e_2, e_3, e_4, e_8\}$$

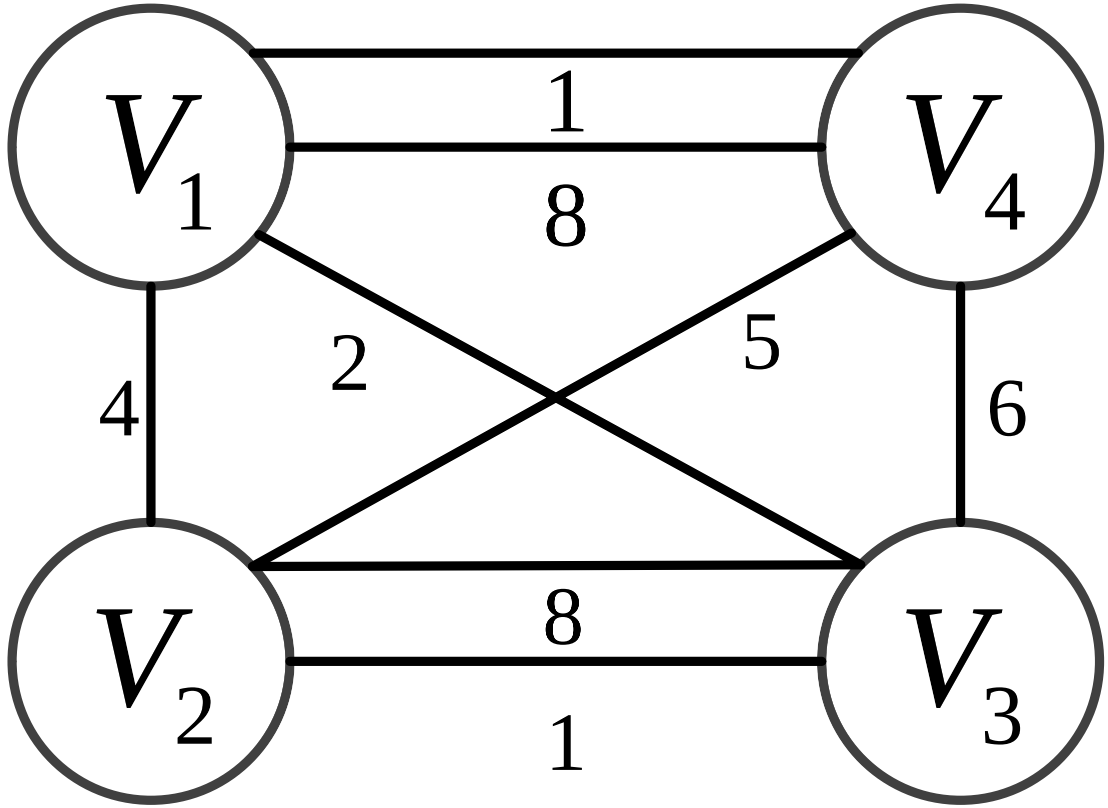
<u>Замечание:</u> Представленный алгоритм Краскала не учитывает веса ребер и используется только для невзвешенных графов.

Построение минимального покрывающего дерева по алгоритму Краскала

Ребра упорядочиваются сначала **по возрастанию** весов и **по нумерации** для ребер с одинаковыми весами.

Для каждого минимального дерева в качестве ответа может быть подсчитан вес дерева $\mathbf{W}_{T_{\min}}$, который складывается из весов всех входящих в дерево ребер.

Пример построения минимального покрывающего дерева по алгоритму Краскала

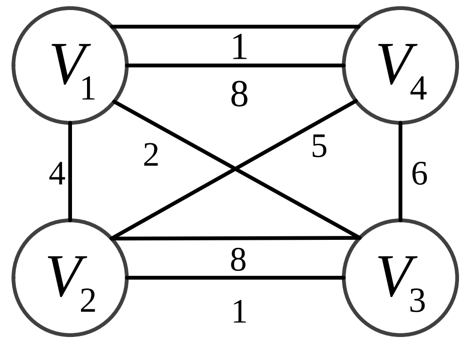


Пример 2. Построить минимальное по весу покрывающее дерево по алгоритму Краскала

Решение для примера 2

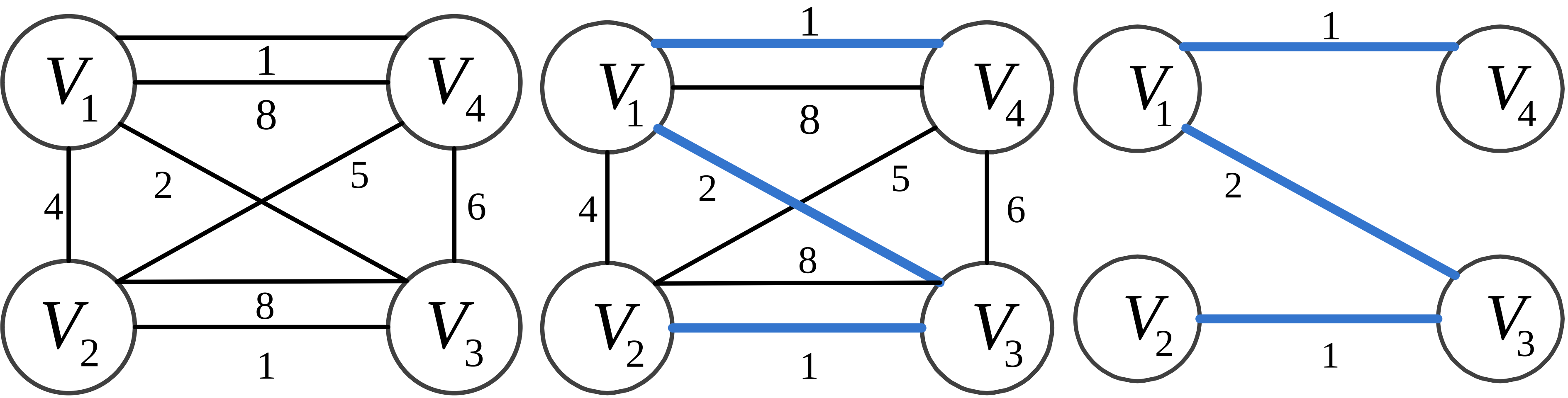
Упорядочим множество ребер **E** по возрастанию весов

$$\mathbf{E} = \{(V_1 \overset{1}{,} V_4),(V_2 \overset{1}{,} V_3),(V_1 \overset{2}{,} V_3),(V_1 \overset{4}{,} V_2),(V_2 \overset{5}{,} V_4),(V_3 \overset{6}{,} V_4),(V_1 \overset{8}{,} V_4),(V_2 \overset{8}{,} V_3)\}$$



Ребро	Цвет	Букет №1	Букет №2
$\varnothing$	–	Пуст	Пуст
$(V_1 \overset{1}{,} V_4)$	Синий (1)	$\{V_1, V_4\}$	Пуст
$(V_2 \overset{2}{,} V_3)$	Синий (2)	$\{V_1, V_4\}$	$\{V_2, V_3\}$
$(V_1 \overset{2}{,} V_3)$	Синий (3)	$\{V_1, V_2, V_3, V_4\}$	Пуст

Ответ для примера 2



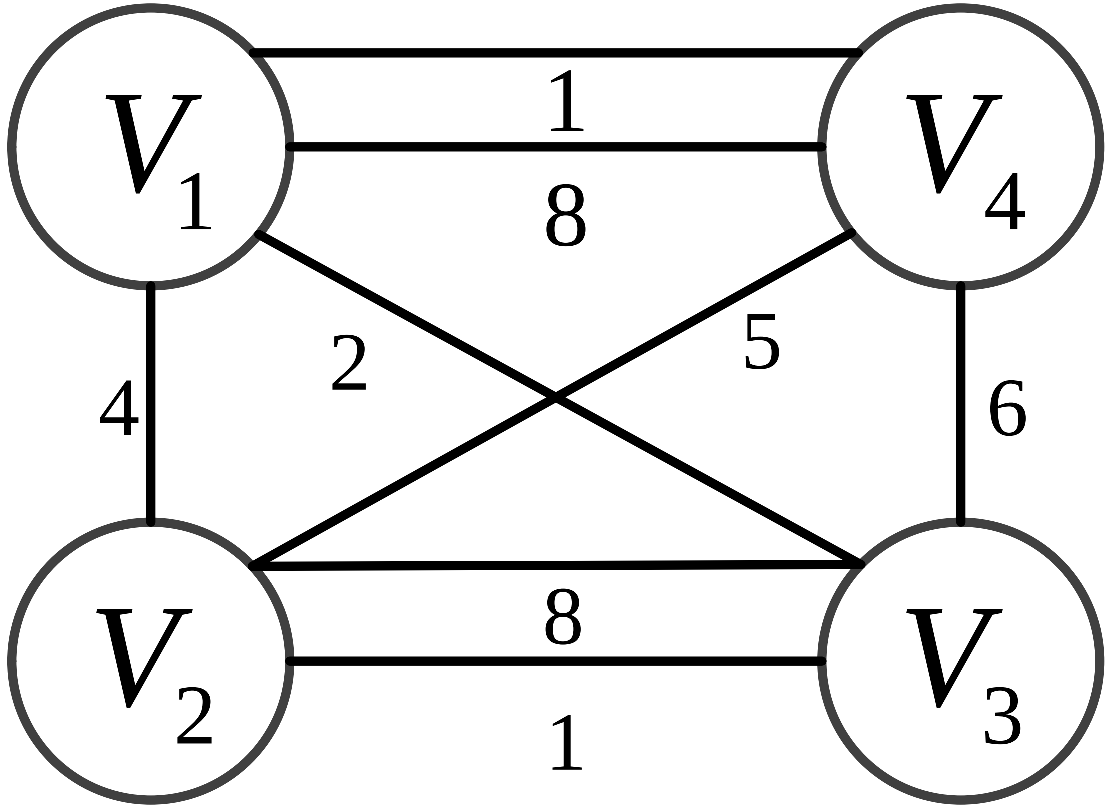
<u>Ответ:</u> Минимальное покрывающее дерево $T = \{(V_1, V_3), (V_1, V_4), (V_2, V_3)\}$ по алгоритму Краскала, вес дерева $\mathbf{W}_{T_{\min}} = 1 + 1 + 2 = 4$.

Построение максимального покрывающего дерева по алгоритму Краскала

Ребра упорядочиваются **по убыванию весов**, а затем **по нумерации** для ребер с одинаковым весом.

Для каждого максимального дерева может быть найден вес дерева $\mathbf{W}_{T_{\max}}$, который складывается из весов всех входящих в дерево ребер.

Пример построения максимального покрывающего дерева по алгоритму Краскала

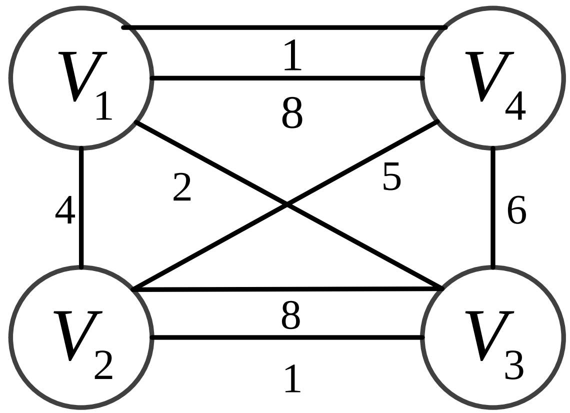


Пример 3. Построить максимальное по весу покрывающее дерево по алгоритму Краскала

Решение для примера 3

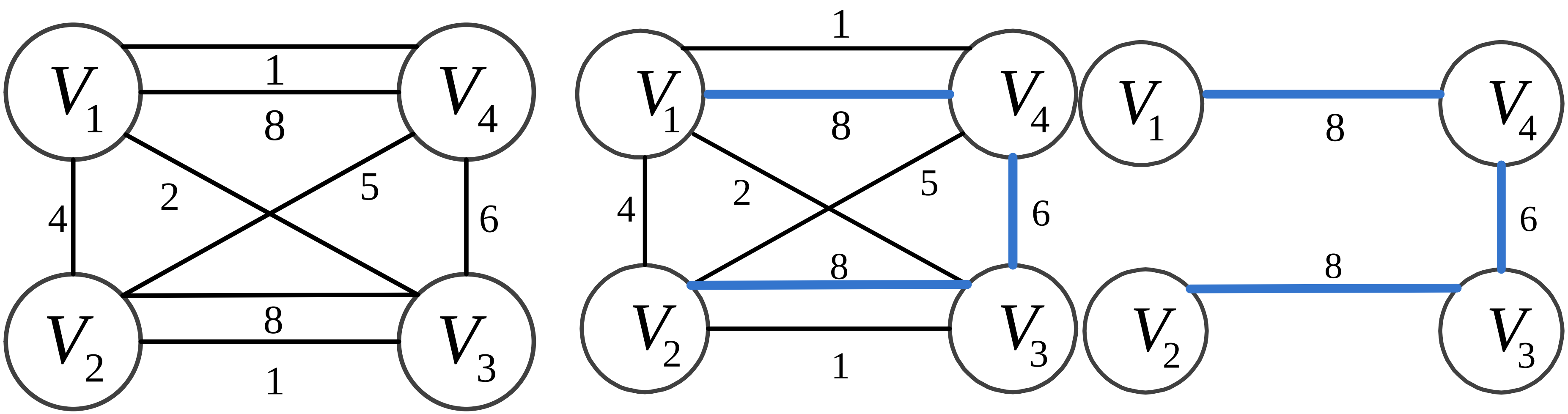
Упорядочим множество ребер $\mathbf{E}$ по убыванию весов:

$$\mathbf{E}=\{(V_1\overset{8}{,}V_4),(V_2\overset{8}{,}V_3),(V_3\overset{6}{,}V_4),(V_2\overset{5}{,}V_4),(V_1\overset{4}{,}V_2),(V_1\overset{2}{,}V_3),(V_1\overset{1}{,}V_4),(V_2\overset{1}{,}V_3)\}.$$



Ребро	Цвет	Букет №1	Букет №2
$\varnothing$	–	Пуст	Пуст
$(V_1\overset{8}{,}V_4)$	Синий (1)	$\{V_1,V_4\}$	Пуст
$(V_2\overset{8}{,}V_3)$	Синий (2)	$\{V_1,V_4\}$	$\{V_2,V_3\}$
$(V_3\overset{6}{,}V_4)$	Синий (3)	$\{V_1,V_2,V_3,V_4\}$	Пуст

Ответ для примера 3



<u>Ответ:</u> Получаем максимальное по весу покрывающее дерево по алгоритму Краскала $T=\{(V_1,V_4),(V_2,V_3),(V_3,V_4)\}$, вес дерева $\mathbf{W}_{T_{\max}}=8+8+6=22$.

Тема следующей лекции:

«Алгоритм Прима»