Теория конечных графов

Алгоритм Прима

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

Алгоритм Прима также называют алгоритмом поиска соседа. Алгоритм применяется только для взвешенных графов. Алгоритм Прима является альтернативным вариантом алгоритму Краскала, позволяет найти минимальное по весу или максимальное по весу покрывающее дерево.

Будем считать, что в данном алгоритме букет будет только один, этот букет будет включать в себя вершины дерева. Вводится дополнительное множество $\mathbf{E}'$ (на начальном этапе совпадающее с упорядоченным множеством ребер $\mathbf{E}$), из которого удаляются рассмотренные ребра.

Построение минимального покрывающего дерева по алгоритму Прима

Начало: Граф связный и взвешенный $G=(\mathbf{V},\mathbf{E})$. Ребра упорядочены по возрастанию весов и по нумерации для ребер с одинаковым весом. Букет не сформирован. Петли удалены. Из кратных ребер оставляем ребро с минимальным весом. При удалении из множества $\mathbf{E}$ петель и кратных ребер получаем множество $\mathbf{E}'$.

Шаг 1. Выбираем из упорядоченного множества $\mathbf{E}'$ первое ребро (V_i,V_j) и включаем его в дерево $T_{\min}$. Обе концевые вершины включаются в букет $\{V_i,V_j\}$. $\mathbf{E}':=\mathbf{E}'\setminus\{(V_i,V_j)\}$.

Построение минимального покрывающего дерева по алгоритму Прима

Шаг 2. Из множества $\mathbf{E}'$ выбираем первое ребро (V_i, V_j), инцидентное какой-либо вершине из сформированного букета. Если $\mathbf{E}' = \varnothing$, то переходим к шагу 3.

После выбора ребра возможны 2 случая:

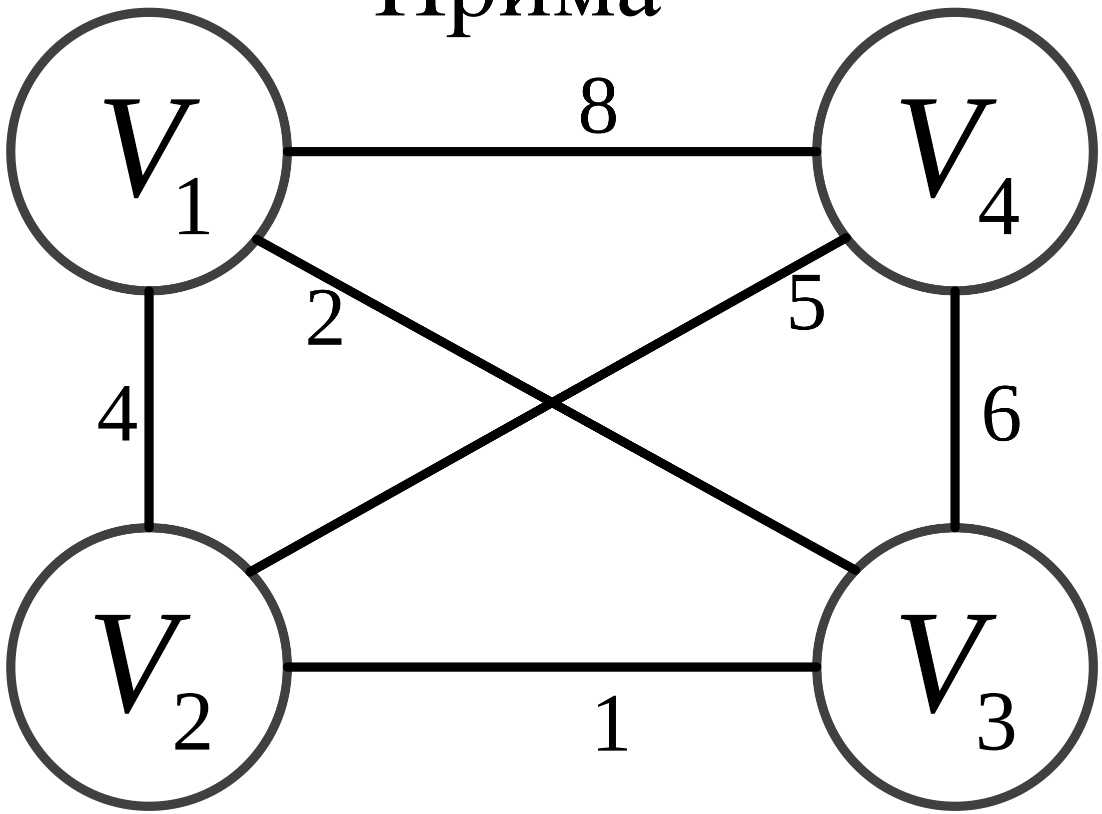
А. Ребро (V_i, V_j) составляет цикл с существующими ребрами дерева. Тогда $\mathbf{E}' := \mathbf{E}' \setminus \left(V_i, V_j\right)$ и возвращаемся к началу шага 2.

Б. Ребро (V_i, V_j) не образует с существующими ребрами дерева цикл. Тогда новая вершина добавляется к вершинам сформированного букета. $\mathbf{E}' := \mathbf{E}' \setminus \left(V_i, V_j\right)$. Ребро добавляется в дерево. Далее возвращаемся к началу шага 2.

Шаг 3. Если дерево (букет) включает в себя все вершины графа, то минимальное покрывающее дерево построено.

Конец алгоритма.

Пример построения минимального покрывающего дерева по алгоритму Прима

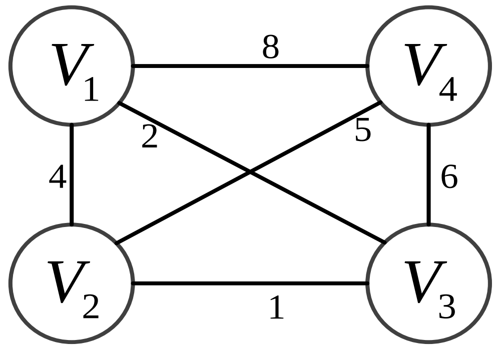


Пример 1. Построить минимальное по весу покрывающее дерево по алгоритму Прима

Решение для примера 1

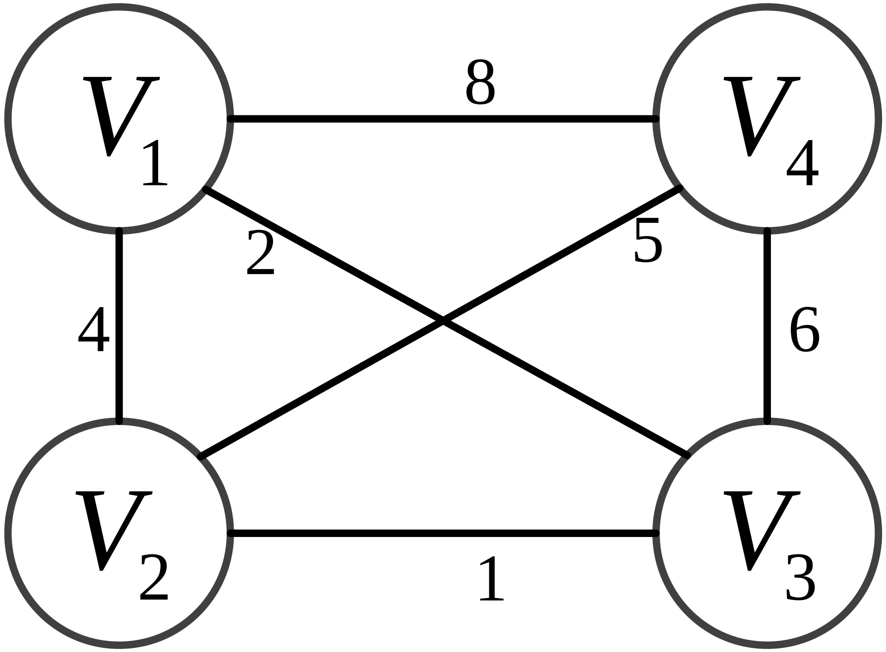
Упорядочим множество ребер $\mathbf{E}$ по возрастанию весов:

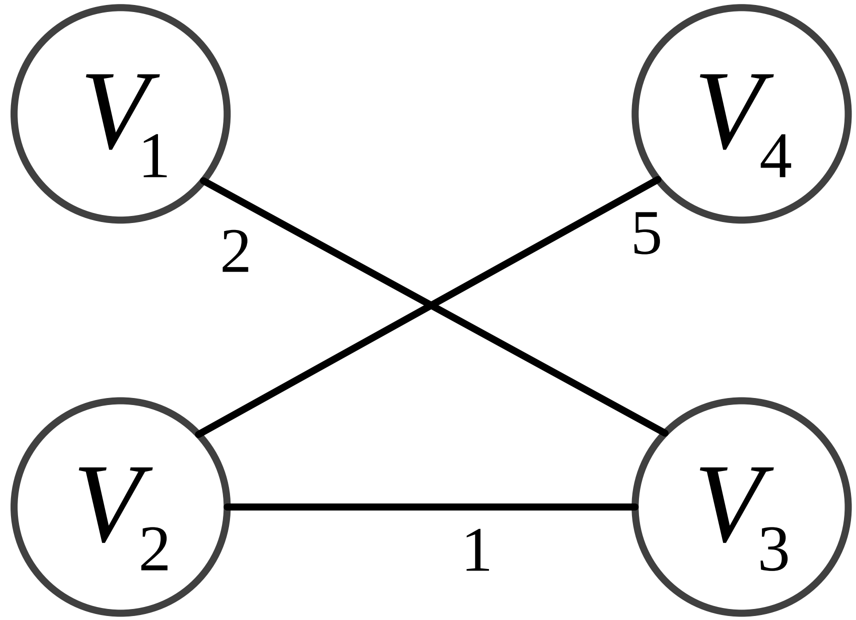
$$\mathbf{E}=\{\overset{1}{(V_2,V_3)},\overset{2}{(V_1,V_3)},\overset{4}{(V_1,V_2)},\overset{5}{(V_2,V_4)},\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}.$$



Ребро	Букет	$\mathbf{E}'$
$\varnothing$	$\varnothing$	$\{\overset{1}{(V_2,V_3)},\overset{2}{(V_1,V_3)},\overset{4}{(V_1,V_2)},\overset{5}{(V_2,V_4)},\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}$
$\overset{1}{(V_2,V_3)}$	$\{V_2,V_3\}$	$\{\overset{2}{(V_1,V_3)},\overset{4}{(V_1,V_2)},\overset{5}{(V_2,V_4)},\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}$
$\overset{2}{(V_1,V_3)}$	$\{V_1,V_2,V_3\}$	$\{\overset{4}{(V_1,V_2)},\overset{5}{(V_2,V_4)},\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}$
$\overset{4}{(V_1,V_2)}$ цикл	$\{V_1,V_2,V_3\}$	$\{\overset{5}{(V_2,V_4)},\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}$
$\overset{5}{(V_2,V_4)}$	$\{V_1,V_2,V_3,V_4\}$	$\{\overset{6}{(V_3,V_4)},\overset{8}{(V_1,V_4)}\}$

Ответ для примера 1





Минимальное по весу покрывающее дерево $T = \{(V_2 \overset{1}{,} V_3), (V_1 \overset{2}{,} V_3), (V_2 \overset{5}{,} V_4)\}$, вес дерева $\mathbf{W}_{T_{\min}} = 1 + 2 + 5 = 8$.

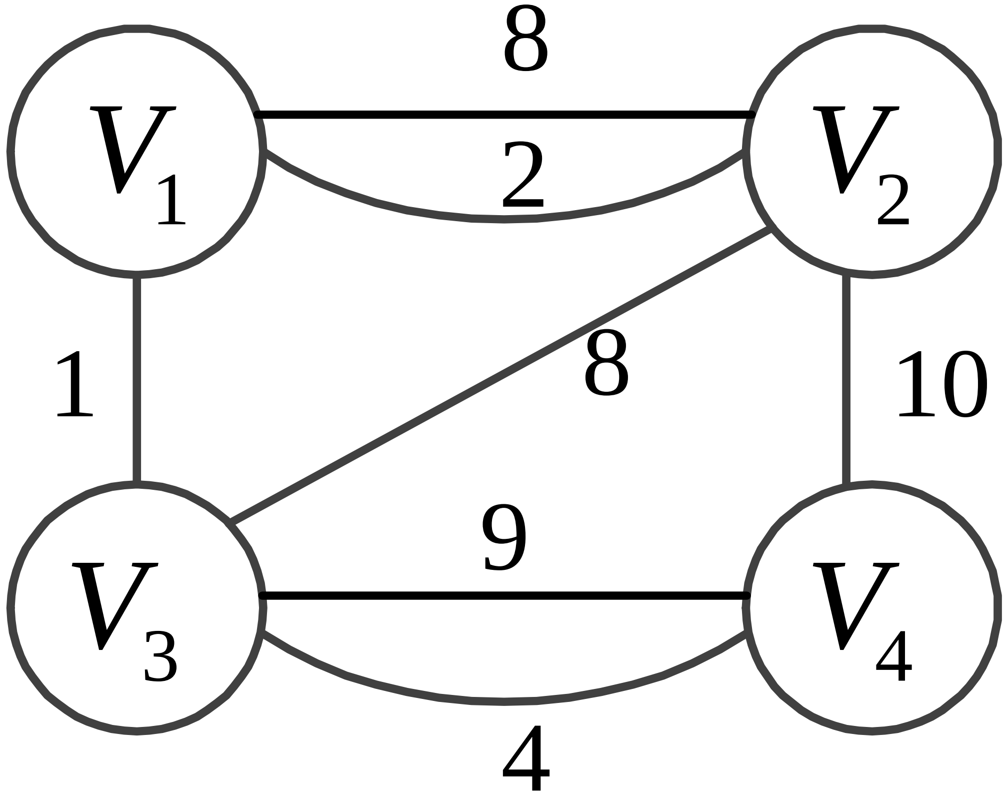
Построение максимального покрывающего дерева по алгоритму Прима

Отличия от построения минимального покрывающего дерева:

1) Ребра упорядочиваются по убыванию весов, а затем по нумерации для ребер с одинаковым весом. Ребро с меньшей нумерацией стоит первым среди ребер с одинаковым весом.

2) Множество $\mathbf{E}'$ получается из множества $\mathbf{E}$ путем исключения петель и кратных ребер (из кратных ребер оставляем ребро с максимальным весом).

Пример построения максимального покрывающего дерева по алгоритму Прима

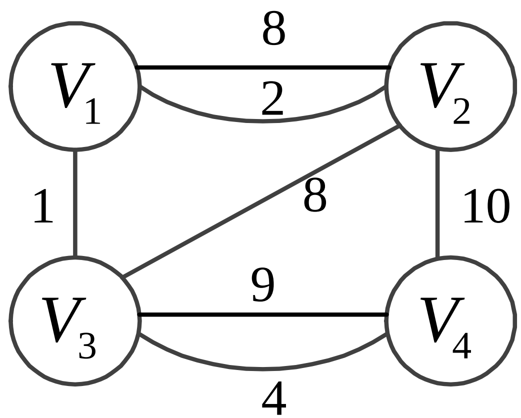


Пример 2. Построить максимальное по весу покрывающее дерево
по алгоритму Прима

Решение для примера 2

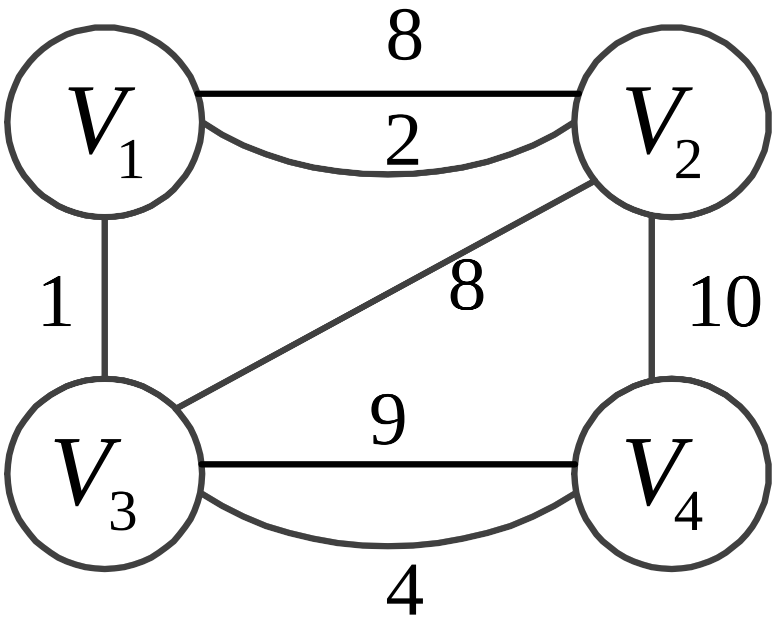
Упорядочим множество ребер $\mathbf{E}$ по убыванию весов:

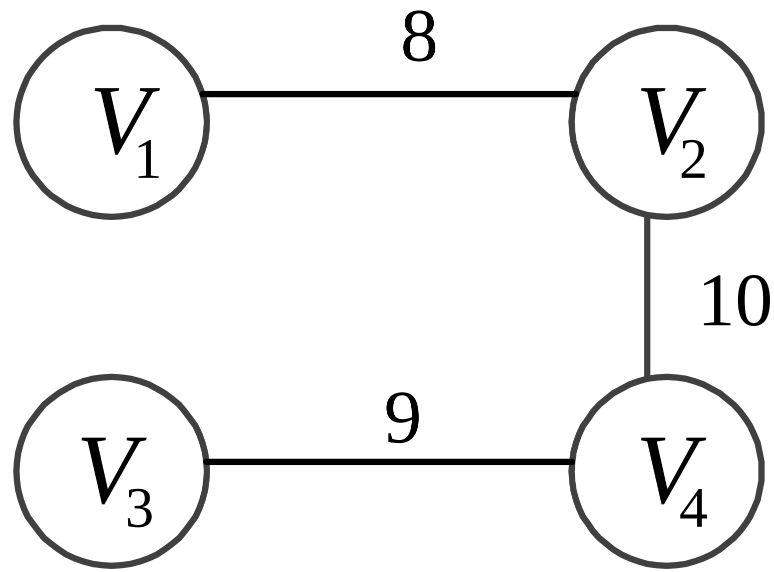
$$\mathbf{E}=\{\overset{10}{(V_2,V_4)},\overset{9}{(V_3,V_4)},\overset{8}{(V_1,V_2)},\overset{8}{(V_2,V_3)},\overset{4}{(V_3,V_4)},\overset{2}{(V_1,V_2)},\overset{1}{(V_1,V_3)}\}\,.$$



Ребро	Букет	$\mathbf{E}'$
$\varnothing$	$\varnothing$	$\{\overset{10}{(V_2,V_4)},\overset{9}{(V_3,V_4)},\overset{8}{(V_1,V_2)},\overset{8}{(V_2,V_3)},\overset{1}{(V_1,V_3)}\}$
$\overset{10}{(V_2,V_4)}$	$\{V_2,V_4\}$	$\{\overset{9}{(V_3,V_4)},\overset{8}{(V_1,V_2)},\overset{8}{(V_2,V_3)},\overset{1}{(V_1,V_3)}\}$
$\overset{9}{(V_3,V_4)}$	$\{V_2,V_3,V_4\}$	$\{\overset{8}{(V_1,V_2)},\overset{8}{(V_2,V_3)},\overset{1}{(V_1,V_3)}\}$
$\overset{8}{(V_1,V_2)}$	$\{V_1,V_2,V_3,V_4\}$	$\{\overset{8}{(V_2,V_3)},\overset{1}{(V_1,V_3)}\}$

Ответ для примера 2





Максимальное по весу покрывающее дерево $T = \{(V_2 \overset{10}{,} V_4), (V_3 \overset{9}{,} V_4), (V_1 \overset{8}{,} V_2)\}$, вес дерева $\mathbf{W}_{T_{\max}} = 10 + 9 + 8 = 27$.

Тема следующей лекции:

«Алгоритм Дейкстры»