**Лабораторная работа 11. Построение минимального остовного дерева.**

**Цель работы:** изучить алгоритмы Краскала и Прима для построения минимального остовного дерева.

### Краткие теоретические сведения

**Остов минимального веса**

*Остов -*это подграф (частичный граф), который может быть построен из графа удалением некоторых ребер и который является деревом.

Рассмотрим следующую задачу: во взвешенном связном графе требуется найти остов минимального веса. Эта задача часто возникает при проектировании, например, линий электропередачи, трубопроводов, дорог и т. п., когда требуется заданные центры соединить некоторой системой каналов связи так, чтобы любые два центра были связаны либо непосредственно соединяющим их каналом, либо через другие центры и каналы и чтобы общая длина (или стоимость) каналов связи была минимальной. В этой ситуации заданные центры можно считать вершинами полного графа с весами ребер, равными длинам (стоимости) соединяющих эти центры каналов. Тогда искомая сеть будет кратчайшим остовным подграфом полного графа. На рисунке показаны различные остовные деревья графа



Рисунок 11.1 Граф и его остовные деревья

Очевидно, что этот кратчайший остовный подграф должен быть деревом. Так как полный граф Kn содержит nn-2 различных остовных деревьев, то решение этой задачи перебором вариантов потребовало бы чрезвычайно больших вычислений даже при малых n. Существуют эффективные алгоритмы, решающие эту задачу, например алгоритм Дж. Краскала (1956 г.) и Р. Прима (1957 г.), применимые к произвольному связному графу.

**Алгоритм Краскала**

Общая формулировка задачи об остове минимального веса (о кратчайшем остове): в связном взвешенном графе G порядка n найти остов минимального веса. Алгоритм Краскала, решающий эту задачу, заключается в следующем.

1. Все ребра графа сортируются по весу.
2. Текущее множество рёбер устанавливается пустым.
3. К множеству вершин последовательно добавляются ребра из отсортированного списка так, чтобы не появлялись циклы.
4. Когда таких ребер больше нет - КОНЕЦ

Затем, пока это возможно, проводится следующая операция: из всех рёбер, добавление которых к уже имеющемуся множеству не вызовет появление в нём цикла, выбирается ребро минимального веса и добавляется к уже имеющемуся множеству. Когда таких рёбер больше нет, алгоритм завершён. Подграф данного графа, содержащий все его вершины и найденное множество рёбер, является его [остовным деревом](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D1%82%D0%BE%D0%B2%D0%BD%D0%BE%D0%B5_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE" \o "Остовное дерево) минимального веса.

Пример

Дан граф



Его матрица смежности имеет вид



Cписок ребер имеет вид

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ребро | V1-V2 | V1-V3 | V2– V3 | V2-V4 | V3-V4 | V3-V5 | V3-V6 | V4-V6 | V5-V6 |
| Вес | 2 | 1 | 3 | 9 | 7 | 6 | 8 | 4 | 5 |

Отсортированный список ребер имеет вид

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ребро | V1-V3 | V1-V2 | V2– V3 | V4-V6 | V5-V6 | V3-V5 | V3-V4 | V3-V6 | V2-V4 |
| Вес | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Порядок добавления ребер для построения остовного дерева минимального веса

Сложность Краскала O(E log V).

**Алгоритм Прима**

Этот алгоритм начинает работу с произвольно выбранной вершины.

1. Выбранная вершина соединяется с ближайшей вершиной. Соединенные вершины образуют связное множество, остальные вершины – несвязное множество.
2. В множество связных вершин добавляется вершина, находящаяся на кратчайшем расстоянии к любой из вершин связного множества.
3. Множества связных и несвязных вершин корректируются,
4. Если вершин больше нет, то КОНЕЦ иначе переход к п.2
5. алгоритм заканчивает работу, когда все вершины попадут в множество связных вершин (рис. 34). Построение остовного дерева начнем с вершины V5.
6. Если веса некоторых ребер совпадают, предпочтение отдается ребру, исходящему из вершины с меньшим номером, а, если ребра исходят из одной и той же вершины, предпочтение отдается ребру, входящему в вершину с меньшим номером.



Алгоритм Прима отличается от алгоритма Краскала только тем, что на каждом шаге строится не просто ациклический граф, а дерево.

Алгоритм Прима можно представить в виде

1. Выбираем ребро e1 ={a, b} минимального веса и строим дерево T1, полагая V(T1) = {a, b}, E(T1) ={e1}.

2. Если дерево Ti порядка i +1 уже построено и i < n-1, то среди ребер, соединяющих вершины этого дерева с вершинами графа G, не входящими в Ti, выбираем ребро ei+1 минимального веса.

3. Строим дерево Ti+1, присоединяя к Ti ребро ei+1 вместе с его не входящим в Ti концом.

1. Порядок выполнения работы
2. Выбрать по номеру студента в журнале свой вариант задания.
3. По списку дуг с указанием их длин составить рисунок ориентированного графа.
4. Найти для этого графа минимальный остов используя алгоритм Краскала (для четных вариантов) или алгоритм Прима (для нечетных вариантов).

**Задания**

Найти минимальный остов графа, заданного списком ребер,.

**1.** (1,2) – 3; (1,3) – 6; (1,4) – 2; (1,5) – 5; (2,3) – 4; (2,4) – 3; (2,5) – 1;

(3,4) – 2; (3,5) – 5; (4,5) – 2.

**2.** (1,2) – 6; (1,6) – 5; (1,8) – 5; (2,3) – 4; (3,4) – 1; (3,8) – 3; (4,5) – 1;

(4,7) – 2; (5,6) – 2; (6,7) – 4; (7,8) – 1.

**3.** (1,2) – 5; (1,3) – 3; (1,6) – 4; (2,3) – 6; (2,6) – 1; (3,4) – 2; (3,5) – 5;

(4,5) – 1; (4,6) – 1; (5,6) – 4.

**4.** (1,2) – 1; (1,3) – 7; (1,6) – 5; (2,3) – 4; (2,6) – 2; (3,4) – 3; (3,5) – 6;

(3,6) – 1; (4,5) – 2; (4,6) – 5; (5,6) – 3.

**5.** (1,2) – 1; (1,8) – 2; (2,3) – 2; (2,4) – 7; (2,8) – 3; (3,4) – 5; (4,5) – 5;

(4,6) – 6; (5,6) – 3; (6,7) – 4; (6,8) – 1; (7,8) – 1.

**6.** (1,2) – 4; (1,3) – 5; (1,4) – 2; (1,5) – 3; (2,3) – 1; (2,4) – 2; (2,5) – 3;

(3,4) – 5; (3,5) – 6; (4,5) – 7.

**7.** (1,2) – 4; (1,6) – 1; (1,8) – 6; (2,3) – 5; (3,4) – 2; (3,8) – 7; (4,5) – 2;

(4,7) – 3; (5,6) – 3; (6,7) – 5; (7,8) – 5.

32

**8.** (1,2) – 1; (1,3) – 7; (1,6) – 5; (2,3) – 4; (2,6) – 2; (3,4) – 3; (3,5) – 6;

(4,5) – 2; (4,6) – 5; (5,6) – 3.

**9.** (1,2) – 5; (1,3) – 3; (1,6) – 4; (2,3) – 6; (2,6) – 1; (3,4) – 2; (3,5) – 5;

(3,6) – 2; (4,5) – 1; (4,6) – 1; (5,6) – 4.

**10.** (1,2) – 5; (1,8) – 1; (2,3) – 1; (2,4) – 3; (2,8) – 2; (3,4) – 1; (4,5) – 4;

(4,6) – 5; (5,6) – 4; (6,7) – 6; (6,8) – 2; (7,8) – 2.

**11.** (1,2) – 5; (1,3) – 3; (1,4) – 7; (1,5) – 2; (2,3) – 2; (2,4) – 1; (2,5) – 2;

(3,4) – 4; (3,5) – 5; (4,5) – 2.

**12.** (1,2) – 5; (1,6) – 2; (1,8) – 5; (2,3) – 3; (3,4) – 1; (3,8) – 2; (4,5) – 7;

(4,7) – 2; (5,6) – 2; (6,7) – 4; (7,8) – 1.

**13.** (1,2) – 2; (1,3) – 2; (1,6) – 4; (2,3) – 5; (2,6) – 1; (3,4) – 2; (3,5) – 5;

(4,5) – 7; (4,6) – 1; (5,6) – 3.

**14.** (1,2) – 3; (1,3) – 1; (1,6) – 4; (2,3) – 7; (2,6) – 2; (3,4) – 5; (3,5) – 4;

(3,6) – 1; (4,5) – 3; (4,6) – 2; (5,6) – 1.

**15.** (1,2) – 3; (1,8) – 2; (2,3) – 3; (2,4) – 1; (2,8) – 5; (3,4) – 2; (4,5) – 4;

(4,6) – 4; (5,6) – 1; (6,7) – 7; (6,8) – 1; (7,8) – 5.

**16.** (1,2) – 2; (1,3) – 4; (1,4) – 1; (1,5) – 1; (2,3) – 3; (2,4) – 5; (2,5) – 2;

(3,4) – 3; (3,5) – 4; (4,5) – 6.

**17.** (1,2) – 2; (1,6) – 3; (1,8) – 4; (2,3) – 4; (3,4) – 5; (3,8) – 6; (4,5) – 1;

(4,7) – 2; (5,6) – 1; (6,7) – 3; (7,8) – 1.

**18.** (1,2) – 3; (1,3) – 6; (1,6) – 3; (2,3) – 2; (2,6) – 5; (3,4) – 1; (3,5) – 4;

(4,5) – 1; (4,6) – 1; (5,6) – 2.

**19.** (1,2) – 2; (1,3) – 2; (1,6) – 4; (2,3) – 5; (2,6) – 1; (3,4) – 2; (3,5) – 5;

(3,6) – 3; (4,5) – 7; (4,6) – 1; (5,6) – 3.

**20.** (1,2) – 2; (1,8) – 1; (2,3) – 7; (2,4) – 2; (2,8) – 2; (3,4) – 1; (4,5) – 4;

(4,6) – 5; (5,6) – 3; (6,7) – 5; (6,8) – 3; (7,8) – 4.

**21.** (1,2) – 7; (1,3) – 1; (1,4) – 3; (1,5) – 5; (2,3) – 3; (2,4) – 2; (2,5) – 2;

(3,4) – 4; (3,5) – 4; (4,5) – 1.

**22.** (1,2) – 7; (1,6) – 3; (1,8) – 4; (2,3) – 1; (3,4) – 2; (3,8) – 1; (4,5) – 3;

(4,7) – 2; (5,6) – 5; (6,7) – 4; (7,8) – 2.

**23.** (1,2) – 3; (1,3) – 1; (1,6) – 4; (2,3) – 7; (2,6) – 2; (3,4) – 5; (3,5) – 4;

(4,5) – 3; (4,6) – 2; (5,6) – 1.

**24.** (1,2) – 2; (1,3) – 7; (1,6) – 4; (2,3) – 1; (2,6) – 3; (3,4) – 5; (3,5) – 6;

(3,6) – 5; (4,5) – 4; (4,6) – 7; (5,6) – 2.

**25.** (1,2) – 2; (1,8) – 3; (2,3) – 4; (2,4) – 7; (2,8) – 5; (3,4) – 7; (4,5) – 4;

(4,6) – 6; (5,6) – 2; (6,7) – 1; (6,8) – 5; (7,8) – 5.

33

**26.** (1,2) – 1; (1,3) – 2; (1,4) – 4; (1,5) – 5; (2,3) – 2; (2,4) – 3; (2,5) – 1;

(3,4) – 4; (3,5) – 6; (4,5) – 7.

**27.** (1,2) – 1; (1,6) – 2; (1,8) – 6; (2,3) – 2; (3,4) – 3; (3,8) – 7; (4,5) – 4;

(4,7) – 1; (5,6) – 5; (6,7) – 4; (7,8) – 7.

**28.** (1,2) – 2; (1,3) – 7; (1,6) – 4; (2,3) – 1; (2,6) – 3; (3,4) – 5; (3,5) – 6;

(4,5) – 4; (4,6) – 7; (5,6) – 2.

**29.** (1,2) – 3; (1,3) – 1; (1,6) – 4; (2,3) – 7; (2,6) – 2; (3,4) – 5; (3,5) – 4;

(3,6) – 5; (4,5) – 3; (4,6) – 2; (5,6) – 1.

**30.** (1,2) – 3; (1,8) – 2; (2,3) – 3; (2,4) – 1; (2,8) – 5; (3,4) – 2; (4,5) – 4;

(4,6) – 4; (5,6) – 1; (6,7) – 7; (6,8) – 5; (7,8) – 5.

Содержание отчета о лабораторной работе

1. Титульный лист
2. Цель работы и задание для выполнения
3. Рисунок графа.
4. Таблицы и рисунки построения миинимального остовного дерева.при добавлении каждого ребра
5. Выводы по лабораторной работе.

Контрольные вопросы

1. Что такое остов минимального веса?
2. Какова сложность алгоритма Краскала?
3. Какова сложность алгоритма Прима?
4. Каковы основные этапы в алгоритме Краскала?
5. Каковы основные этапы в алгоритме Прима?