Федеральное государственное образовательное учреждение

Высшего профессионального образования

Национальный исследовательский технологический университет «МИСиС»

Институт информационных технологий и компьютерных наук (ИТКН)

Кафедра автоматизированных систем управления (АСУ)

Курсовая работа

по курсу «Алгоритмы дискретной математики»

Тема:

«Алгоритм Краскала»

Выполнил: студент 2 курса,

гр. БИВТ-21-9 Тен Л.П.

Руководитель: Валова А.А.

МОСКВА 2023

Оглавление

**Введение3**

**Постановка задачи4**

**Теоретическая часть4**

Описание алгоритма5

Основная идея5

Асимптотическая оценка5

Сравнение алгоритма с другими6

Примеры работы7

**Заключение9**

**Список литературы9**

**Программные приложения10**

**Введение**

Алгоритмы дискретной математики играют существенную роль в различных областях информатики и компьютерных наук. В современном информационном обществе, где данные являются одним из самых ценных ресурсов, разработка и эффективное использование алгоритма становится ключевой задачей. Алгоритмы представляют собой систему формальных методов и инструментов, использующихся для решения разнообразных задач, связанных с ограниченными и дискретными структурами данных. Идея любого метода основывается на теории графов, комбинаторике, логике и других разделах дискретной математики.  
Целью данной курсовой работы является изучение и исследование конкретного алгоритма дискретной математики, используемого для решения задач, требующих дискретного анализа и обработки данных. Работа направлена на изучение основной идеи, теоретическое и практическое применение алгоритма.  
В рамках работы будет проведен анализ сложности алгоритма, с целью оценки его эффективности и оптимальности, а также его реализация на языке программирования и сравнение c другими методами, решающими подобные задачи.

**Постановка задачи**

Формальная постановка задачи:

Вначале текущее множество рёбер устанавливается пустым. 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 минимального веса.

Неформальная постановка задачи:

Допустим, у нас есть граф без рёбер, нам нужно каким-то образом соединить все вершины между собой так, чтобы можно было попасть из любой вершины в другую, не имея при этом циклов в получившемся графе с минимально возможной суммой весов включенных рёбер.

**Теоретическая часть**

**Описание**

Алгоритм Краскала является одним из алгоритмов для нахождения минимального остовного дерева. Подобные алгоритмы применимы в различных областях, начиная от кластеризации данных до построения компьютерных, транспортных сетей. Данный алгоритм используется для взвешенных связных неориентированных графов. Алгоритм впервые был описан Джозефом Крускалом в 1956 году.

Минимальным остовным деревом связного взвешенного графа называется его связный подграф, состоящий из всех вершин исходного дерева и некоторых его ребер, причем сумма весов ребер минимально возможная.

Алгоритм Краскала – это алгоритм для построения минимального остовного дерева в связном взвешенном неориентированном графе. Минимальное остовное дерево представляет собой подграф, содержащий все вершины исходного графа и являющийся деревом с наименьшей суммой весов ребер.

Пусть у нас есть N вершин, переменная X[i,j] будет представлять ребро, связывающее i-ую и j-ую вершины, Y – вес ребра.

При этом x[i,j] примет значение 1, если ребро включено в покрывающее дерево и 0 − в противном случае.

Целевая функция: Z(x) = Y1X[i,j] + Y2X[i,j] + Y3X[i,j] + Y4X[i,j] + … + Y(N-1)X[i,j]

Ограничения задачи представлены следующей системой неравенств:

{

X[i,j] + X[i,j + 1] >= 1

X[i,j] + X[i,j +1] + X[i+1,j+2] >= 1

X[i,j] + X[i,j + 1] + X[i,j] + X[i,j +1] + X[i+1,j+2] = N-1

}

X[i,j] >= 0 ; X[i,j] ∈{0;1};

Первые два ограничения определяют свойство смежности вершин связного графа, третье ограничение определяет свойство дерева, а именно число ребер дерева на единицу меньше числа вершин, остальные два свойства определяют условие не отрицательности переменных и условие включения ребра в дерево.

**Основная идея**

Основная идея алгоритма Краскала будет описана в следующих шагах:

1. Необходимо отсортировать все ребра исходного графа по неубыванию в порядке возрастания весов.
2. Создать пустой список, который будет представлять наше пустое остовное дерево, которое будет заполняться ребрами из исходного графа по ходу алгоритма.
3. Пройтись по отсортированным ребрам и добавить их в остовное дерево, если это не создаст цикл. Добавление ребра не создаст цикл, если его ребро, которое добавляется в дерево связывает две вершины из разных компонент связности.
4. Повторить шаг 3 до тех пор, пока не будут добавлены все вершины в остовное дерево или все ребра будут рассмотрены.

Псевдокод можно посмотреть по ссылке ниже:

<https://docs.google.com/document/d/1SKaNtp4Z7HuQJ86HXM_8BZ8yKIpTC7tO41qhWH5oa6Q/edit>

**Асимптотическая оценка**

Алгоритм Краскала обладает временной сложностью O(E log V), где Е – количество ребер в графе, V – количество вершин в графе. Это делает его эффективным для работы с большими графами.

Sort() – O(E logV)

makeSet() – O(V)

find() – O(1)

union() – O(1)

**Сравнение алгоритма с другими**

Существует еще один алгоритм, для построения минимального остовного дерева – это алгоритм Прима.

Основная его суть заключается в следующих шагах:

1. Выбрать произвольную вершину в графе и добавить ее в минимальный остов.
2. Пометить выбранную вершину как посещенную и найти все ребра, связанные с этой вершиной.
3. Выбрать ребро с наименьшим весом из найденных ребер, которое соединяет посещенную и непосещенную вершины.
4. Добавить выбранное ребро в минимальное остовное дерево и пометить связанную с ним вершину как посещенную.
5. Повторять шаги 3 и 4 до тех пор, пока не будут посещены все вершины графа.

Как можно заметить, в отличие от алгоритма Краскала, алгоритм Прима строит минимальное остовное дерево постепенно, добавляя ребра одно за другим, начиная с одной начальной вершины.

Алгоритм Прима в основном сфокусирован на вершинах, ищет ближайшие непосещенные вершины и соответствующие им ребра.

В итоге имеем следующее:

1. Алгоритм Прима начинает с одной начальной вершины, в то время как алгоритм Краскала начинает с сортировки всех ребер
2. В алгоритме Прима процесс выбора ребер основан на вершинах и построении дерева из начальной вершины, в то время как алгоритм Краскала основывается на рассмотрении всех ребер в порядке неубывания их весов.
3. Алгоритм Прима обычно более эффективен для плотных графов, тогда как алгоритм Краскала лучше работает на разреженных графах

**Примеры работы**

Таблица 1. Примеры работы алгоритма

|  |  |  |
| --- | --- | --- |
| Исходный граф | Ожидаемый результат | Фактический результат |
|  | 15 | 15 |
|  | 2 | 2 |
|  | 2 | 2 |
|  | 2 | 2 |
|  | 0 | 0 |

Приведены примеры работы алгоритма Краскала, где:

1. Обычный неориентированный связный граф
2. Граф с двумя вершинами и одним ребром
3. Граф с двумя вершинами и двумя ребрами
4. Граф с тремя вершинами и тремя ребрами одинакового веса
5. Граф с одной вершиной и одним ребром

**Заключение**

В заключение, алгоритм Краскала представляет собой эффективный и простой в реализации метод решения задачи минимального остовного дерева. Он основывается на принципе «объединяй и властвуй», который позволяет найти минимальное остовное дерево в связном графе, объединяя подграфы постепенно и добавляя ребра с минимальными весами. Алгоритм Краскала обладает рядом преимуществ, таких как линейная сложность по времени, возможность работы с неполными графами и возможность поиска минимальных остовных деревьев в неориентированных и ориентированных графах.

Он находит широкое применение в различных областях, таких как сетевое планирование, транспортная логистика, телекоммуникации и многих других. Благодаря своей простоте и эффективности, алгоритм Краскала является одним из наиболее популярных методов решения задачи минимального остовного дерева.

В итоге, алгоритм Краскала является мощным инструментом, который позволяет эффективно решать задачу минимального остовного дерева. Его простота и эффективность делают его незаменимым инструментом для решения широкого спектра задач в различных областях.

**Список литературы**

1. Расул Юнусов Алгоритм Краскала, Прима для нахождения минимального остовного дерева: [Электронный ресурс]. URL: <https://habr.com/ru/articles/569444/>. (Дата обращения: 15.05.2023)
2. Минимальное остовное дерево. Алгоритм Крускала с системой непересекающихся множеств: [Электронный ресурс]. URL: <https://e-maxx.ru/algo/mst_kruskal_with_dsu>. (Дата обращения: 15.05.2023)

**Программные приложения**

<https://replit.com/@LeonidTen/KraskalAlgoADM#Main.java>