ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО

ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**«**Казанский (Приволжский) Федеральный Университет**»**

Институт информационных технологий и интеллектуальных систем

Построение минимального остовного дерева. Алгоритм Краскала.

Выполнил:

Хайруллин Рустам Айратович

Преподователь:

Гафурова Полина Олеговна

Казань - 2021

Алгоритм Краскала.

Алгоритм Краскала — эффективный алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа.

Алгоритм описан Джозефом Краскалом в 1956 году, этот алгоритм почти не отличается от алгоритма Борувки, предложенного Отакаром Борувкой в 1926 году.

Принцип работы:

В начале 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.

Причём для корректности работы алгоритма изначально множество ребер должно быть отсортировано.

Шаги для реализации алгоритма Краскала следующие:

* Сортировать все ребра от малого веса до высокого.
* Возьмите ребро с наименьшим весом и добавьте его в остовное дерево. Если добавление ребра создало цикл, то отклоните это ребро.
* Продолжайте добавлять ребра, пока не достигнете всех вершин.

Оценка скорости алгоритма.

До начала работы алгоритма необходимо отсортировать рёбра по весу, это требует O(E × log(E)) времени. После чего компоненты связности удобно хранить в виде системы непересекающихся множеств. Все операции в таком случае займут O(E × α(E, V)), где α — функция, обратная к функции Аккермана. Поскольку для любых практических задач α(E, V) < 5, то можно принять её за константу, таким образом, общее время работы алгоритма Краскала можно принять за O(E \* log(E)).

Плюсы и минусы

* Требуется предварительная сортировка для корректной работы алгоритма.
* Действительно находит минмальное остовное дерево за довольно эффективное время - O(E \* log(E)).

График зависимости времени от количества итераций:

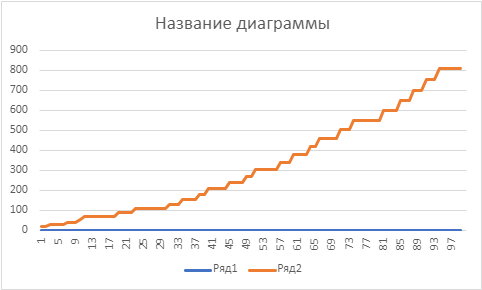


График зависимости времени от количества вершин в графе:

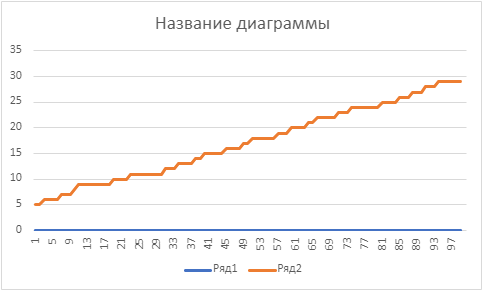
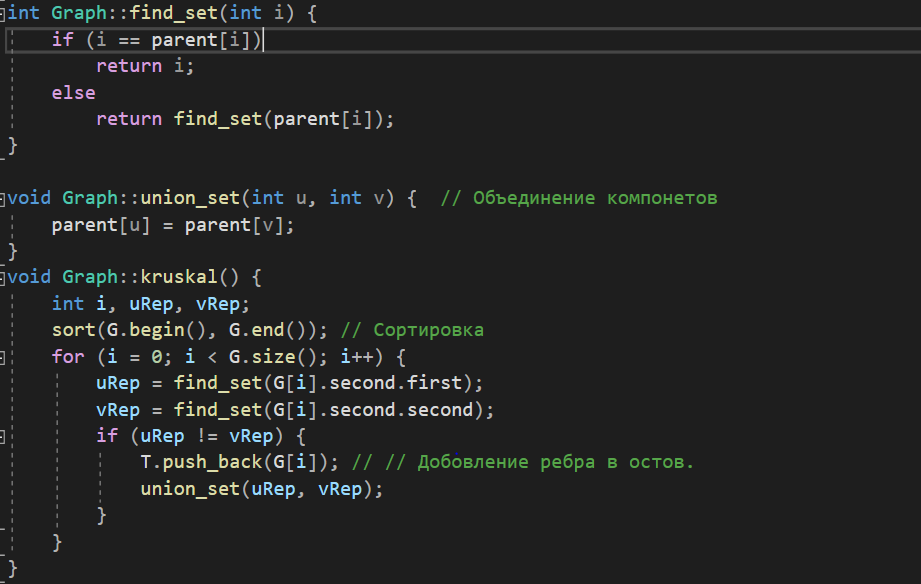


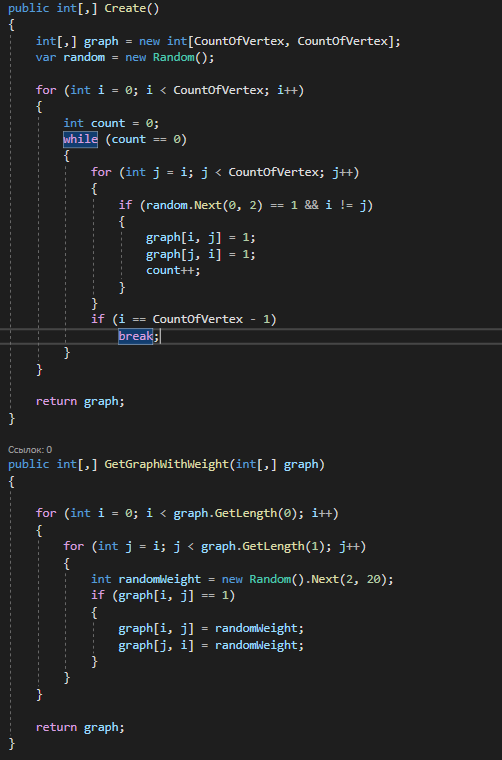
График зависимости количества итераций от количества вершин в графе:

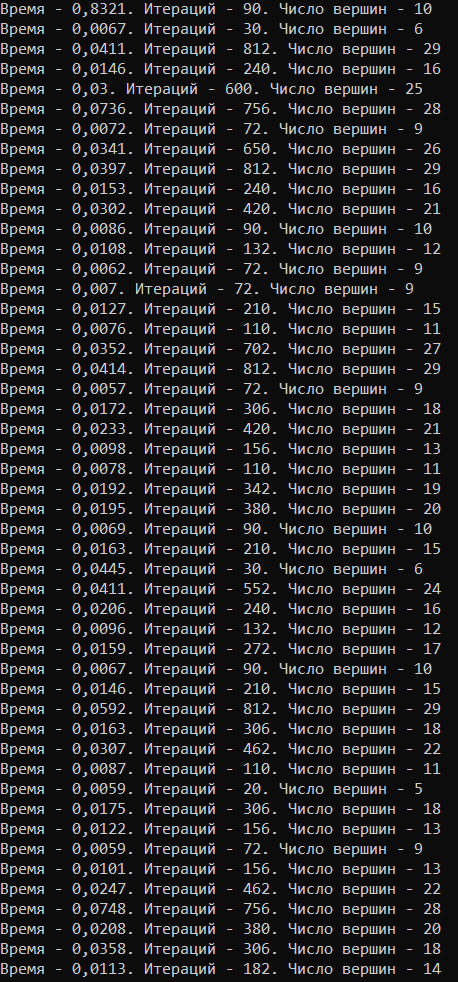


Фрагмент кода:



Генератор взвешенных графов:





Заключение:

Без сомнений, алгоритм Краскала по нахождению минимального остовного дерева является эффективным алгоритмом, который используется во многих IT сферах: анализ данных, графика и даже звук.

Литература:

* <https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9A%D1%80%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D0%B0#%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0>
* <https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9A%D1%80%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D0%B0#.D0.A0.D0.B5.D0.B0.D0.BB.D0.B8.D0.B7.D0.B0.D1.86.D0.B8.D1.8F>
* http://espressocode.top/difference-between-prims-and-kruskals-algorithm-for-mst/