1 задача: Минимальное остовное дерево (2)

3 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

Дан неориентированный связный граф. Требуется найти вес минимального остовного дерева в этом графе.
Вариант 1. С помощью алгоритма Прима.
Вариант 2. С помощью алгоритма Крускала.
Вариант 3. С помощью алгоритма Борувки.
Ваш номер варианта прописан в ведомости.

Формат входного файла.

Первая строка содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \le n \le 20000$, $0 \le m \le 100000$).
Следующие m строк содержат описание ребер по одному на строке.
Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \le b_i, e_i \le n$, $0 \le w_i \le 100000$).

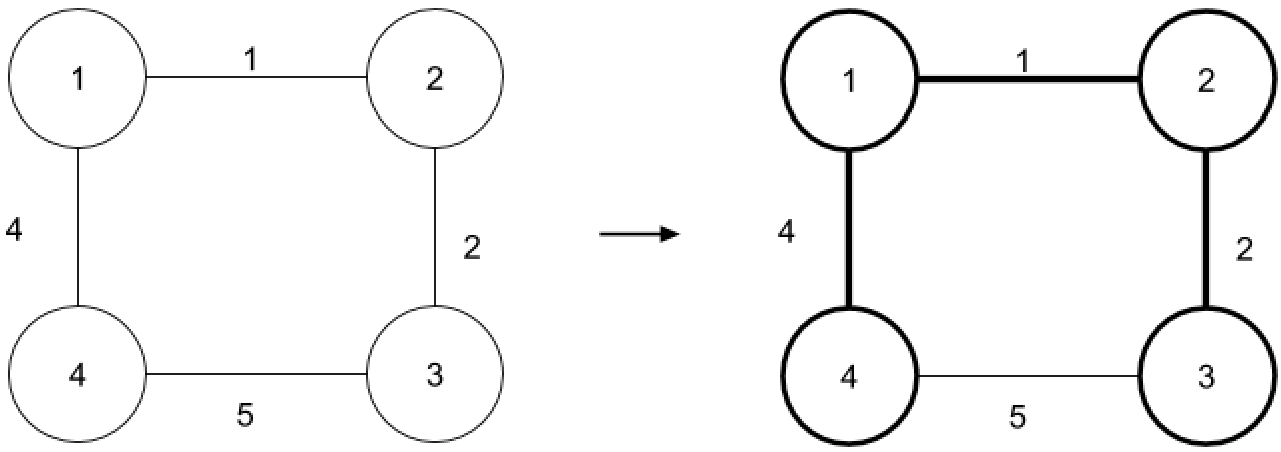
Формат выходного файла.

Выведите единственное целое число - вес минимального остовного дерева.

Пример:

in	out
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Иллюстрация примера:



2. Описание алгоритма

Все рёбра сортируются по возрастанию. Каждая вершина помещается в своё множество. Затем в порядке возрастания каждое ребро (u, v), $u \in U$, $v \in V$ проверяется на то, что u и v находятся в разных множествах, т. е. $U \neq V$. Если это так, то (u, v) является безопасным и помещается в MST, множества U и V объединяются. Проверяем рёбра пока все вершины не окажутся в одном множестве

3. Доказательство корректности

Доказательство смотри в Кормене на в главе 23.2

4. Время работы и дополнительная память

Время работы зависит от реализации структуры данных для хранения множеств

$T = O(E \log V + V^2)$ - наивная реализация (в контест не заходит)

$T = O(E \log V)$ - структура данных “система непересекающихся множеств”

$M = O(V + E)$

5. Доказательство времени работы

Сортировка рёбер требует $O(E \log V)$ (так как $E < V^2$, $\log E = O(\log V)$)

<u>Наивная реализация множеств:</u> храним вектор int, в котором хранится номер множества, в котором хранится каждая вершина. Инициализация V множеств - $O(1)$. Проверка того, в каком множестве находится вершина, занимает $O(1)$, объединение множеств занимает $O(V)$. Всего происходит E проверок и V объединений, поэтому в сумме обработка всех рёбер занимает $O(V^2 + E)$. В итоге получаем время $O(E \log V + V^2 + V + E) = O(E \log V + V^2)$

<u>Умная реализация:</u> выполняется $O(E)$ операций поиска рёбер и объединения множеств, вместе с V операциями инициализации множеств уходит $O((V + E)\alpha(V))$, где α - обратная функция *Аккермана*. Так как граф связный, $E \geq V - 1$, поэтому $O((V + E)\alpha(V)) = O(E\alpha(V))$, $\alpha(V) < 5$, поэтому получаем $O(E)$. В итоге получаем время $O(E \log V + E) = O(E \log V)$