

Задание 1

Алгоритм ALG_1

Для сортировки рёбер по весу необходимо $\Theta(|E| \log |E|)$ операций
(в частном случае, если веса - ограниченное подмножество целых чисел фиксированного размера (в битах), можно применить radix-sort за $\Theta(|E| + K)$)
Внешний цикл будет выполняться за $\Theta(|E|)$, проверка на связность - обходом в глубину / ширину (dfs / bfs) (корректно, т.к. граф неориентированный)
На каждой итерации обход делается за $\Theta(|V| + |E'|)$, где E' - текущее количество рёбер в графе
На каждой итерации в худшем случае $|E'| = |E|$, если ни одно ребро на предыдущих итерациях не удалялось
Таким образом, цикл работает за $\mathcal{O}(|E| (|E| + |V|))$
Удаление ребра делается за $\mathcal{O}(1)$
Тогда весь алгоритм работает за $\mathcal{O}(|E| \cdot (|E| + |V|) + |E| \log |E|) = \mathcal{O}(|E| \cdot (|E| + |V|))$

Алгоритм ALG_2

Случайную последовательность ребёр графа $G = (V, E)$ можно сгенерировать за $\Theta(|E|)$, сгенерировав последовательность индексов от 0 до $|E| - 1$ за $\Theta(|E|)$ и перемешать её при помощи `std::shuffle` за $\Theta(|E|)$
Внешний цикл выполняется за $\Theta(|E|)$
Проверка на наличие циклов в неориентированном графе - dfs за $\mathcal{O}(|V| + |E|)$
Добавление ребра - за $\mathcal{O}(1)$
Тогда весь алгоритм работает за $= \mathcal{O}(|E| \cdot (|E| + |V|))$