

SET 2

A1.

1. Реализации представлены в приложенном файле.

Для которого алгоритма используется вектор на ребер, в Alg1 и Alg3 для оптимизации работы DFS используется матрица смежности (вектор списков).

Оценки:

Alg1: # сортировка: $O(E \log E)$

Проверка на связность: $O(V+E)$

Количество проверок: E

Сложность удаления: $O(E)$

Итого: $O(E(V+E))$

Alg2: Проверка на циклы: $O(L(E))$

Кол-во проверок: E

Итого: $O(E L(E))$

Alg3: Проверка на циклы (с поиском макс. ребра): $O(E \cdot N)$
Кол-во проверок: E .
Итого: $O(E^2(N+1))$

~~то сложность~~

общая сложность добавлений и удалений ребер из векторов не превосходит итоговую сложность в данном алгоритме.

2. Alg1:

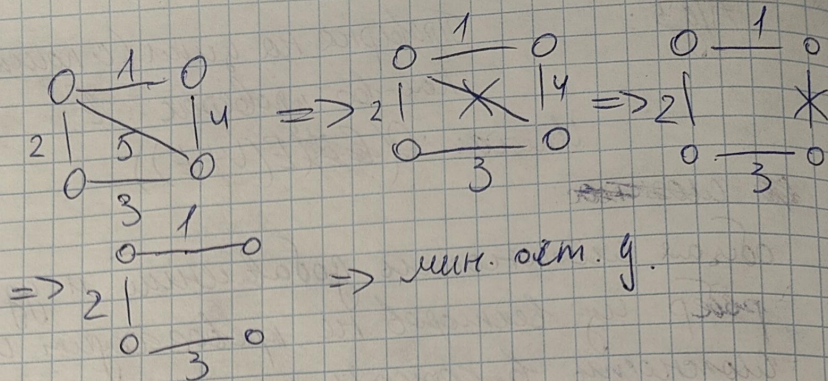
Сначала берется полный граф, затем производится попытка удалить ребро с максимальным возможным весом. В итоге работы получается ~~дерево~~ (если удалить на одно ребро больше, граф будет несвязным), при этом веса ребер минимально возможные. \Rightarrow получили мин. ост. ~~дерево~~.

Alg2: Добавили ребра, чтобы не было циклов.

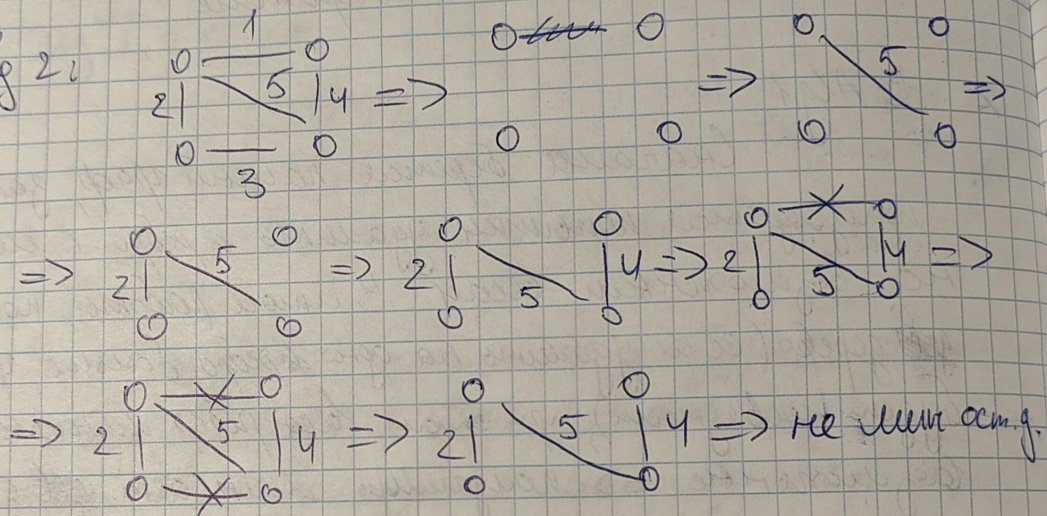
В итоге получили дерево, но оно не обязательно будет минимальным, т.к. при добавлении не учитываются веса. \Rightarrow не мин. ост. д.

Alg3. Соединяем с Alg2, но теперь учитываются и веса при встрече с циклом, удаляем ребро максимального веса. Так переберем все ребра. В итоге во всех получившихся циклах остались минимальные ребра, граф без циклов \Rightarrow получили мин. ост. д.

Alg 1:



Alg 2:



Alg 3:

