**Топологическая сортировка графа**

указать такой линейный порядок на его вершинах, чтобы любое ребро вело от вершины с меньшим номером к вершине с большим номером.

Сначала пробовал хранить граф списком смежности, но по памяти не проходит. Пришел к выводу хранить матрицей смежности, где 1 - есть связь, 0 - нет связи.

graph[count\_vertex \* start + end] - для просмотра связи от start+1 до end+1(вершины).

**Добавление в граф:**

graph[count\_vertex \* start + end] = 1 - добавилили связь от start+1 до end+1.

used - массив, в котором хранятся данные о том, что мы сделали с вершиной.

1 - были уже в этой вершине, если мы зайдем в нее еще раз, то топологическая сортировка невозможна, т.к присутствует цикл. 2 - добавили в out.

**Про dfs(Обход в глубину с добавлением вершин без связи):**

Идем по графу в глубину, если из вершины больше нет необработанных связей, то ложим в out и ключ 2 в used.

Вопросы и ответы:

1) Что вернёт алгоритм если есть несвязные подмножества вершин?  (несвязные - есть рёбра между вершинами множества А и между вершинами множества B, но нет рёбер из A в B и из B в A).  
2) Какова оценка количества операций для данного алгоритма?

1)Работает также, обрабатывая все вершины, если множества упорядочены по номерам, то сначала выведет результат работы topsort(линейный порядок от вершины с меньшим номером к большему) для одного множества, далее таким же образом для второго.

2) O(M) на добавление. Сам алгоритм: сложность равна сложности обхода в глубина(в основе сортировки он и лежит), O(N+M) мы обходим все ребра для каждой вершины(M ребер и N вершин).