**Билет №3**

**Поиск мостов в графе. Простой способ поиска. Использование дополнительных массивов для быстрого поиска мостов, оценка времени работы.**

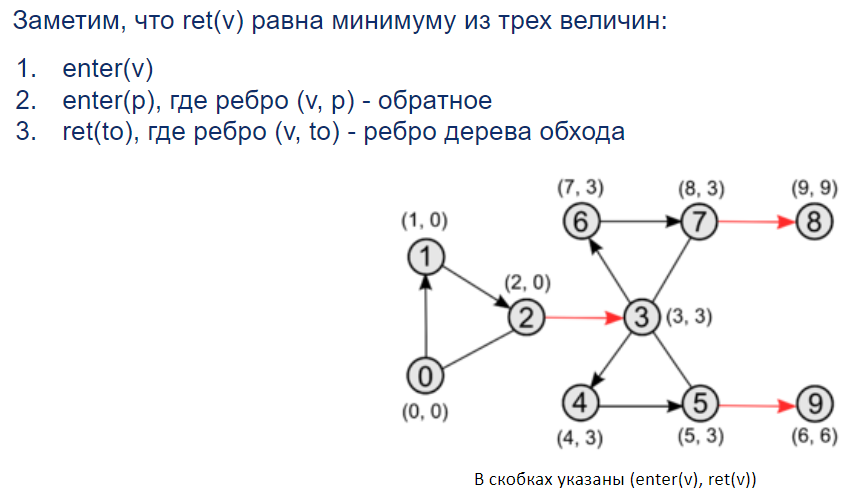
**Задача**: дан неориентированный граф, требуется найти все мосты в нем.

**Простой способ решения**: создаём копию нашего графа и по-очереди удаляем ребра, проверяя остался ли граф связным (как вариант запускаем dfs с какой-нибудь вершины и вводим HashSet used, при посещении вершины добавляем её туда, если после выполнения функции размер used будет равен кол-ву вершин, то граф связен, если нет, то удаленное ребро является мостом)

Анализ: 1) если мы ищем ребро мост, то это только ребро дерево (доказать на рисунке). 2)также заметим, что ребро между точками V и U является мостом тогда и только тогда, когда из вершины U и ее потомков нельзя вернуться в вершину V или подняться выше нее. (пример – в.6 и в.7). Для данной реализации нам помогут функции:

* enter(v) – время входа в вершину v
* ret(v) – минимальное время вхождения из v в потомков v (потомок – это вершина куда можно попасть из v)

Пример: вершины 2 и 3 => ret(2) = 3 те из 3 можно попасть в 2 за время = 2

3) 

4) Самое главное условие : ret(v) > enter(u). Показать на примере вершин 2,3: ret(3) = 2 (время возвращения в вершину 2), enter(2) = 2 => 2>2 => ребро 2, 3 не является мостом.

Приступим к самой задаче

Изображение выглядит как текст

Автоматически созданное описание

**Решение:** <https://pastebin.com/LuxesuYn>

**Время выполнения**: O(n+m)