**Билет№4**

**Поиск точек сочленения. Простой способ поиска. Использование дополнительных массивов для быстрого поиска точек сочленения, оценка времени работы.**

**Задача:** найти все точки сочленения графа.

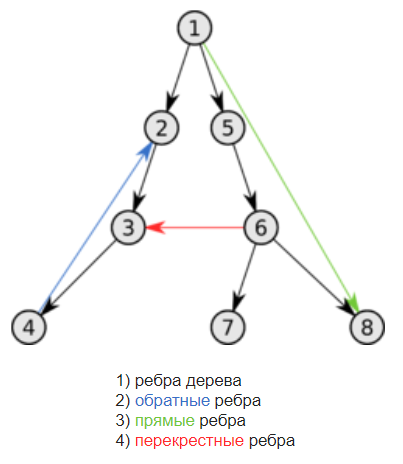
**Простой способ решения:** создаём копию нашего графа и по-очереди удаляем вершины, проверяя остался ли граф связным (как вариант запускаем dfs с какой-нибудь вершины и вводим HashSet used, при посещении вершины добавляем её туда, если после выполнения функции размер used будет равен кол-ву вершин,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

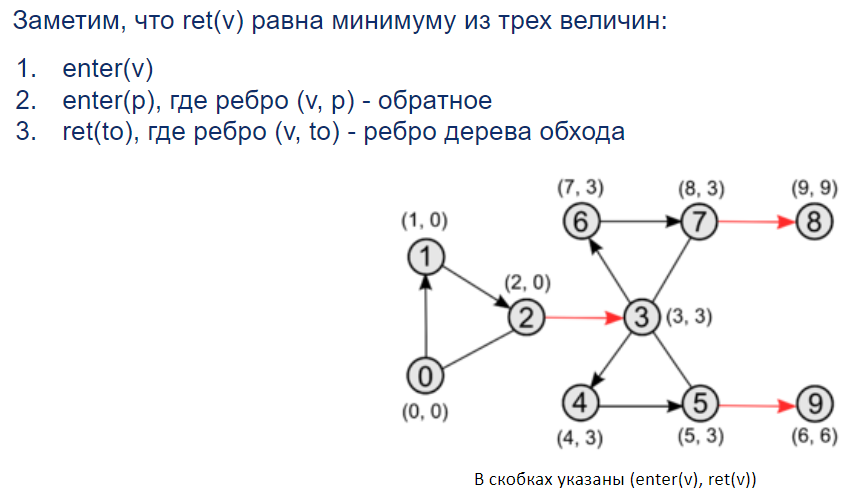
**Анализ:**

1. 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



1. 
2. некорневая вершина u будет являться точкой сочленения, если найдется ребро дерева обхода (u,v) такое, что ret(v) >= enter(u). Те все аналогично прошлой задаче, только со знаком “ >= ”.
3. Корневая вершина будет являться точкой сочленения, если она имеет более одного сына.

Изображение выглядит как текст

Автоматически созданное описание

**Решение:** <https://pastebin.com/3c4T0M53>

**Время выполнения**: O(n+m)