**Спецификация подпрограммы определения ветвей**

**Задача:**

Составить подпрограмму, которая будет создавать и обновлять массив ветвей на основе уже созданного массива зажатых зажимов, массива узлов, массива не узлов (массив, включающий все остальные зажатые зажимы, помимо узлов). Создание/обновление должно происходить при каждом создании/удалении провода.

**Уточненная постановка задачи:**

Имеется три массива:

- massive\_row\_column\_clamped\_clamps,

- massive\_row\_column\_nodes,

- massive\_row\_column\_no\_nodes,

содержащие пары чисел, обозначающих ряд и столбец в поле зажатых зажимов. При этом massive\_row\_column\_clamped\_clamps содержит все зажатые зажимы, то есть к которым подключены элементы. massive\_row\_column\_nodes содержит все зажимы, которые являются узлами, то есть к которым подключены сразу три и более элемента. massive\_row\_column\_no\_nodes содержит зажимы, к которым подключены один или два элемента.

Точно известно, что массивы верные, а элементы могут подключаться только на соседние зажимы.

Опционно может быть раздроблен массив massive\_row\_column\_no\_nodes, на массив соединений (зажимы, в которые подключаются два элемента) и массив неподключенных концов (зажимы, в которые подключается один элемент.

Необходимо точно продумать логичную систему хранения информции о ветви и ее зажимах, чтобы обновление путем добавления или удаления проводов происходило корректно.

Основные случаи:

1. Создан провод на пустом рабочем пространстве
2. К неподключенному концу добавлен провод, конец которого не подключен
3. К соединению добавлен провод, конец которого не подключен
4. К узлу добавлен провод, конец которого не подключен
5. Одновременное 2-4

Ветвь должна иметь возможность создаваться, делиться, дробиться, обновляться

*Основной процесс работы reload\_branches\_when\_creating\_wire:*

При создании провода, он проходит через фильтр:

Определяется сначала тип начального зажима: есть ли у проводов, которые уже были до создания нового (проход по массиву WIRES с абсолютным исключением нового провода) координаты clamp\_start или координаты clamp\_end, совпадающие с начальным зажимом нового провода.

Если тип начального зажима пустой, то определяем тип конечного зажима: Если тип конечного зажима пустой: то создаём ветвь, которая состоит из этого провода

**Случай Стартовая: пустой; Конечная: пустой рассмотрен**

Если тип конечного зажима неподключенный конец, то добавляем провод в ветвь, к которой относится конец

**Случай Стартовая: пустой; Конечная: неподключенный конец рассмотрен**

Если тип конечного зажима соединение, то разделить ветвь, к которой подключается провод на две дочерние: Конечным зажимом нового провода делимая ветвь разделилась на две дочерние ветви, одна из которых останется старой, а вторая будет новой. Для этого создаём приоритетность:

1) У дочерней ветви, к которой относится зажим начала делимой ветви приоритет 1

2) У дочерней ветви, к которой относится зажим конца делимой ветви приоритет 0

3) Если на дочерней ветви есть провод со стрелкой - то это добавляет +2 к приоритету

Таким образом, по наибольшему приоритету выбирается, которая из дочерних ветвей останется под именем делимой ветви, а какая станет новой Случай Стартовая: пустой; Конечная: соединение рассмотрен Если конечный зажим узел, то завершаем создание ветви

**Случай Стартовая: пустой; Конечная: узел рассмотрен**

Если начальный зажим неподключенный конец: добавляем провод в ветвь, которой принадлежит провод с совпавшим зажимом. Далее определяем тип конечного зажима: совершаем проход по массиву WIRES с абсолютным исключением нового провода, для определения, есть ли среди проводов координаты clamp\_start или координаты clamp\_end, совпадающие с конечным зажимом нового провода. Если тип зажима пустой, то просто добавляем провод в ветвь, к концу которой подключается провод

**Случай Стартовая: неподключенный конец; Конечная: пустой рассмотрен**

Если конечный зажим неподключенный конец, то определяем ветвь, которой принадлежит провод, один из концов которых замкнут на данный зажим. Так как смыкаются две ветви в одну, необходимо определить, какая из них исчезнет, а какая останется. Для этого создаём приоритетность:

1) у ветви, на неподключенный конец которой включается начало нового провода приоритет 1

2) у ветви, на неподключенный конец которой включается конец нового провода приоритет 0

3) если у ветви есть значение (и направление) тока, то это добавляет +2 к приоритету Таким образом, по наибольшему приоритету выбирается, какая ветвь останется. Она обновляется и включает теперь все провода прошлой ветви. Слабая ветвь полностью удаляется

**Случай Стартовая: неподключенный конец; Конечная: неподключенный конец рассмотрен**

Если конечный зажим соединение, то без проблем завершаем создание ветви, включающей новый провод. Далее разбираемся с той ветвью, к которой подключился провод. Конечным зажимом нового провода делимая ветвь разделилась на две дочерние ветви, одна из которых останется старой, а вторая будет новой. Для этого создаём приоритетность:

1) У дочерней ветви, к которой относится зажим начала делимой ветви приоритет 1

2) У дочерней ветви, к которой относится зажим конца делимой ветви приоритет 0

3) Если на дочерней ветви есть провод со стрелкой - то это добавляет +2 к приоритету

Таким образом, по наибольшему приоритету выбирается, которая из дочерних ветвей останется под именем делимой ветви, а какая станет новой ветвью

**Случай Стартовая: неподключенный конец; Конечная: соединение рассмотрен**

Если конечный зажим узел, то просто завершаем создание ветви

**Случай Стартовая: неподключенный конец; Конечная: узел рассмотрен**

Если начальный зажим соединение, то ветвь, к которой подключается провод, делится на две ветви. Для этого создаём приоритетность:

1) У дочерней ветви, к которой относится зажим начала делимой ветви приоритет 1

2) У дочерней ветви, к которой относится зажим конца делимой ветви приоритет 0

3) Если на дочерней ветви есть провод со стрелкой - то это добавляет +2 к приоритету

Далее рассматриваем конец нового провода Если конечный зажим пустой, то завершаем создание ветви

**Случай Стартовая: соединение; Конечная: пустой рассмотрен**

Если конечный зажим неподключенный конец, то добавляем провод в ветвь, которой принадлежит неподключенный конец. Случай Стартовая: соединение; Конечная: неподключенный конец рассмотрен. Если конечный зажим соединение, то ветвь, к которой подключается провод, делится на две ветви. Для этого создаём приоритетность:

1) У дочерней ветви, к которой относится зажим начала делимой ветви приоритет 1

2) У дочерней ветви, к которой относится зажим конца делимой ветви приоритет 0

3) Если на дочерней ветви есть провод со стрелкой - то это добавляет +2 к приоритету

Таким образом отрабатывается деление ветви, у которой подключается новый провод. Сам новый провод становится новой ветвью

**Случай Стартовая: соединение; Конечная: соединение рассмотрен**

Если конечный зажим узел, то просто завершаем создание ветви

**Случай Стартовая: соединение; Конечная: узел рассмотрен**

Если начальный зажим узел, то сразу переходим к определению конечного зажима: Если конечный зажим пустой, то завершить создание ветви

**Случай Стартовая: узел; Конечная: пустой рассмотрен**

Если конечный зажим неподключенный конец, то добавить новый провод в ветвь с неподключенным концом и завершить добавление провода

**Случай Стартовая: узел; Конечная: неподключенный конец рассмотрен**

Если конечный зажим соединение, то провести разделение подключаемой ветви согласно регламенту, сам новый провод превращаем в ветвь

**Случай Стартовая: узел; Конечная: соединение рассмотрен**

Если конечный зажим узел, то завершить создание ветви из одного нового провода

*Основной процесс работы reload\_branches\_when\_deleting\_wire:*