Формулировка задачи. Дан взвешенный граф, в котором веса присвоены ребрам. Необходимо найти минимальное остовное дерево имеющую своим корнем одну из вершин графа.

Идея алгоритма. Искомые ребра соединяют вершины. Поэтому возможны две стратегии построения. Можно идти от вершин и для каждой из них искать минимальное ребро (как это сделано в алгоритме Прима) а можно для каждого ребра выяснять можно ли его включить в строящееся дерево. Алгоритм Краскалапредлагает делать это следующим образом. Во-первых, ребра графа пронумеровываем в порядке возрастания весов. Затем для каждого ребра начиная с первого проверяем соединяет или нет оно две несвязные вершины, если да, то его можно включить в остовное дерево. Ясно, что если мы имеем V вершин, то работа алгоритма начинается с V несвязных компонент графа (пока из графа все ребра исключаем). Для того, чтобы их связать необходимо найти V-1 ребро.

Другими словами, алгоритм организует процесс роста компонент связности в процессе которого он объединяются друг с другом до тех пор пока не останется одна являющаяся конечным результатом.

Структуры данных. В алгоритме используется термин «компонента связности». Такой конструкции нет ни в одном языке программирования. Это исключительно математический термин, поэтому для конкретной реализации алгоритма необходимо тщательно продумать вопрос о представлении «компоненты связности» существующими языковыми конструкциями.

Алгоритм

- Создаем список ребер по возрастанию.
- Создаем множество компонент связности каждая из которых содержит ровно одну вершину.
- Пока компонент связности больше чем одна.
    - Взять ребро из начала списка ребер.
    - Если ребро соединяет две разных компоненты связности то
        - Компоненты связности объединить в одну.