Вершины графа представляются списком структур Vertex. У каждой структуры есть указатель на список смежности. Элементами этого списка являются структуры Edge, которые состоят из полей. Если граф ориентирован то . Получается небольшое дублирование данных, но это позволяет использовать одни и те алгоритмы для ориентированного и неориентированного графа.

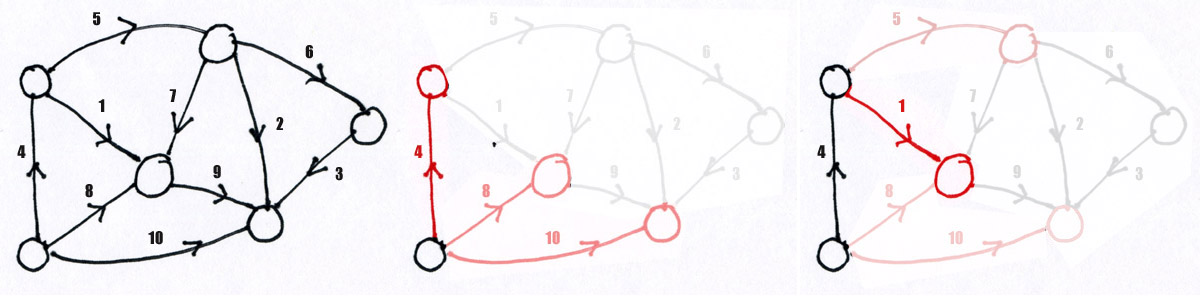
Алгоритм Прима-Ярника

Этот алгоритм обычно называется алгоритмом Прима, так как он стал широко известен по статье Роберта Прима, опубликованной в 1956 г. (левое фото). Впоследствии стало известно, что в 1930 г. этот алгоритм был открыт чешским математиком Войтехом Ярником (1897-1970) (правое фото), основные научные интересы которого были в области теории чисел.

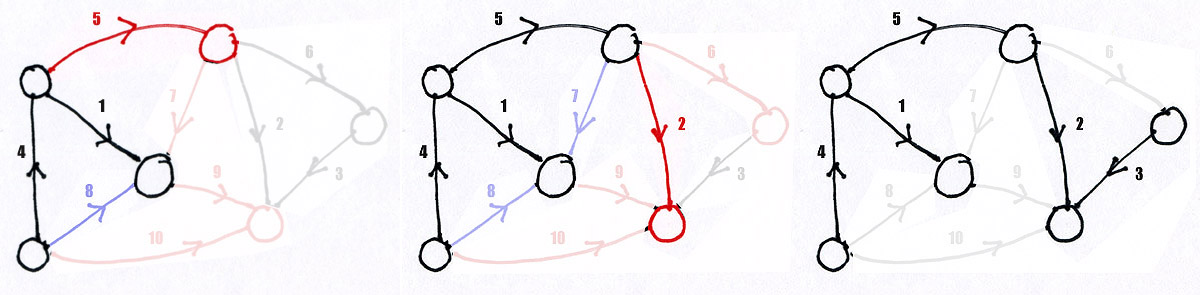
Предложенный ими алгоритм очень прост, для его описания хватит пары строк:

Дерево строится так, как описано в доказательстве теоремы об остовном дереве, но на каждом шаге для пополнения дерева из границы выбирается дуга "с наименьшим весом".

На этих шести рисунках показаны исходный граф, несколько итераций метода и окончательное решение.



На рисунках, соответствующих итерациям черным показано частичное дерево, розовым цветом граница, а красным — лучшая дуга границы и добавляемая вершина. Голубоватым нарисованы дуги, соединяющие вершины дерева и в дерево не входящие.



В этом примере не видно трудностей, связанных с перестройкой границы на каждой итерации. Для преодоления этих трудностей предпринимались специальные модификации метода.

Алгоритм Краскала

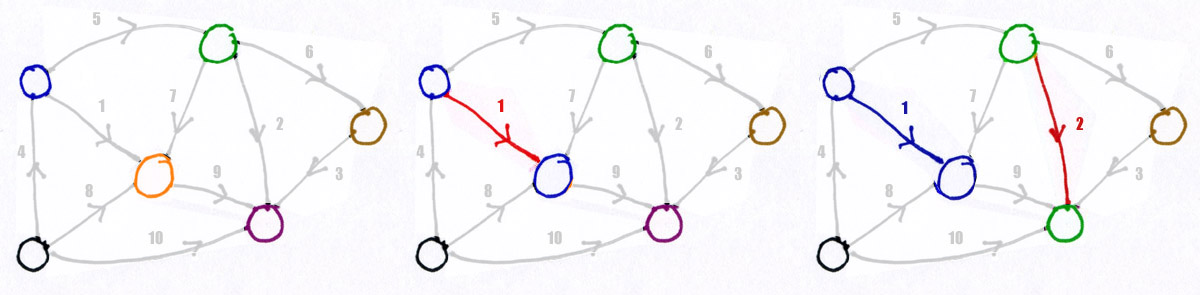
Алгоритм, более привлекательный в вычислительном отношении, был предложен Джозефом Краскалом в 1957 г.

Алгоритм состоит из двух фаз. На подготовительной фазе все дуги удаляются из дерева и упорядочиваются по возрастанию их весов. В графе остаются только вершины, каждая из которых образует отдельную компоненту связности.

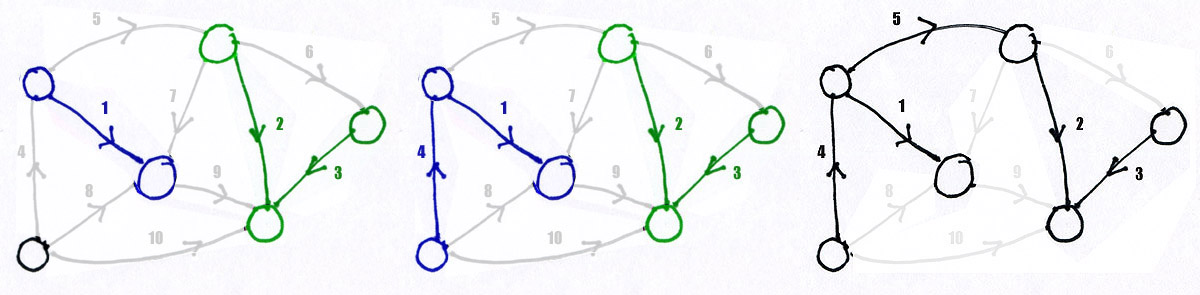
Во второй фазе дуги перебираются в порядке возрастания веса. Если начало и конец очередной дуги принадлежат одной и той же компоненте связности, дуга игнорируется. Если же они лежат в разных компонентах связности. дуга добавляется к графу, а эти две компоненты связности объединяются в одну.

Пример

Алгоритм Краскала мы рассмотрим на том же графе. Начинаем с графа, в котором все дуги удалены, каждая вершина — отдельная компонента связности.



Первая дуга соединяет две различные компоненты связности, они объединяются в одну. То же происходит и со второй дугой. Третья и четвертая дуги также включаются в граф, который становится связным, и остальные дуги даже не рассматриваются.



В этом примере нам ни разу не встретился случай, когда очередную дугу следует пропустить, так как она совединяет две вершины из одной компоненты связности. Представьте себе, что после дуги 2 нам попалась бы дуга 6. Мы бы ее просто игнорировали.