

Постановка задачи. Дан произвольный неориентированный граф. Найти между двумя заданными точками k – кратчайших путей.

Идея. Алгоритм предполагает, что один из кратчайших путей уже найден каким-либо способом. Ясно, что любой другой путь должен отличаться от уже найденного хотя бы одним ребром. Тогда построим новый граф который получим из исходного выбросив одно из ребер найденного кратчайшего пути и уже в новом графе осуществим поиск кратчайшего.

Алгоритм

- Найти первый КРАТЧАЙШИЙ ПУТЬ.
- Записать КРАТЧАЙШИЙ ПУТЬ в список кратчайших
- Выполнять $k - 1$ раз
 - min = Количество ребер исходного графа
 - НОВЫЙ ПУТЬ = пустому
 - Для каждого ребра КРАТЧАЙШЕГО ПУТИ выполнить
 - Построить новый граф полученный из исходного удалением текущего ребра.
 - Найти кратчайший путь в построенном графе.
 - Если длина найденного пути $\leq min$ то min =длине найденного пути и НОВЫЙ ПУТЬ = найденному кратчайшему и УДАЛЕННОЕ РЕБРО = текущее ребро
 - Восстановить исходный граф добавлением текущего ребра
 - Записать НОВЫЙ ПУТЬ в список кратчайших
 - Удалить из графа УДАЛЕННОЕ РЕБРО
 - КРАТЧАЙШИЙ ПУТЬ = НОВЫЙ ПУТЬ

Примечание. Обратите внимание! При переходе на новый шаг внешнего цикла из графа выкидывается ребро именуемое УДАЛЕННЫМ РЕБРОМ. Если этого не делать, то алгоритм может обнаружить уже найденные кратчайшие.