**Алгоритм Диница**

Алгоритм Диница - полиномиальный алгоритм для вычисления максимального потока в транспортной сети. Его придумал израильтянин, когда-то советский программист, Ефим (Хаим) А.Диниц. Этот алгоритм является улучшенной версией алгоритма Эдмондса-Карпа. Его алгоритм справляется с работой за  O(V^2 E) и использует самые короткие пути увеличения. Заполучить эту оценку позволяет введение понятий вспомогательной сети и блокирующего потока.

Введём 3 определения, на которых основан алгоритм Диница. Каждое из этих определений не зависит от других.

**Остаточная сеть** **G^R** строится относительно сети G и некоторому потоку f в этой сети. В итоге построения получается сеть в которой каждому ребру  (u,v) \in G с пропускной способностью c_{uv} и потоком  f_{uv} соответствуют два ребра:

1.1 (u,v) с пропускной способностью c_{uv}^R = c_{uv} - f_{uv}

1.2 (v,u) с пропускной способностью c_{vu}^R = f_{uv}

Здесь можно заметить, что в остаточной сети появятся кратные рёбра, если до построения сеть G содержала как (u,v)  так и (v,u) .

Остаточное ребро несет в себе информацию о том, насколько ещё можно увеличить поток по этому ребру. Если по ребру  (u,v) с пропускной способностью c_{uv} проходит поток  f_{uv} , то оно способно вместить ещё  c_{uv}-f_{uv} единиц потока, а в обратную сторону может пропустить до  f_{uv} единиц потока, и это будет значить отмену потока в начальном направлении.

**Блокирующий поток** в исходной сети это таковой поток, что любой путь из истока s в сток t имеет насыщенное этим потоком ребро. Другими словами, в предоставленной сети никак не найдётся такого пути из истока в сток, вдоль которого позволительно увеличить поток.

Однако блокирующий поток не всегда является максимальным. Об этом и говориться в теореме Форда-Фалкерсона. Поток будет максимальным тогда и лишь тогда, когда в остаточной сети никак не будет пути из истока в сток.

**Слоистая сеть** для данной сети строится следующим образом. Первым делом определяются все длины кротчайших путей из истока ко всем остальным вершинам и обозначаются как \operatorname{dist}(v) . Далее в слоистую сеть добавляют все те рёбра (u,v) исходной сети, для которых справедливо dist(u)=dist(v)+1.

**Алгоритм**

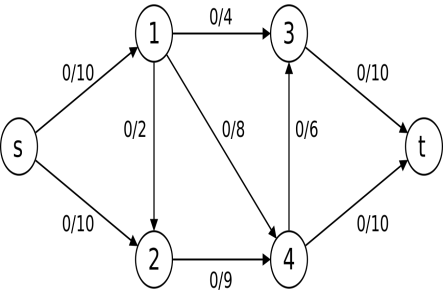
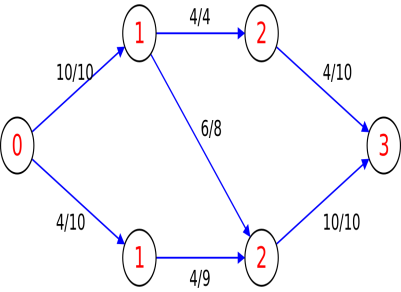
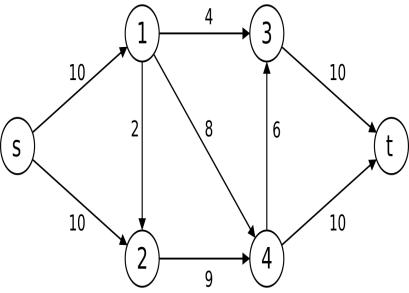
Алгоритм состоит из нескольких шагов. На вход алгоритму подаётся исходная сеть, в которой нужно найти максимальный поток. На выход алгоритм возвращает максимальный поток данной сети.

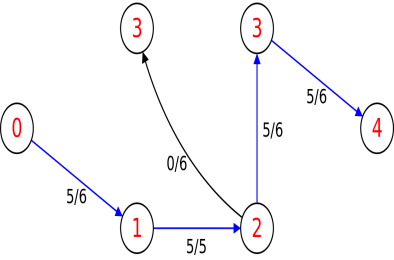
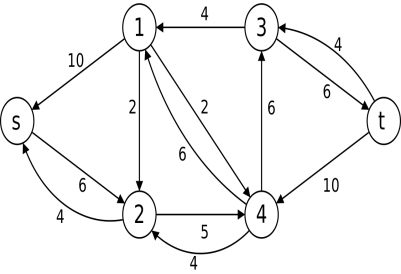
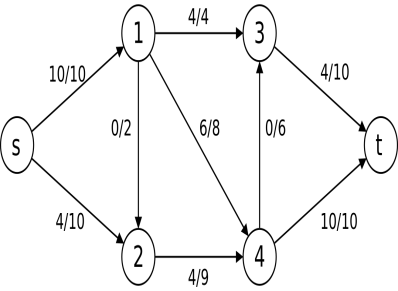
1. Строиться слоистая сеть, если в слоистой сети dist(t)=∞, алгоритм останавливается и выводит максимальный поток.
2. По отношения к слоистой сети строится блокирующий поток.
3. Основной поток дополняется блокирующим.

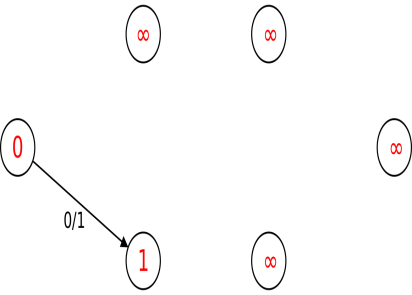
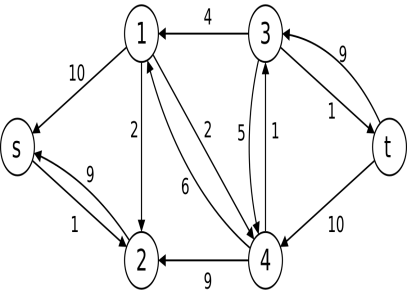
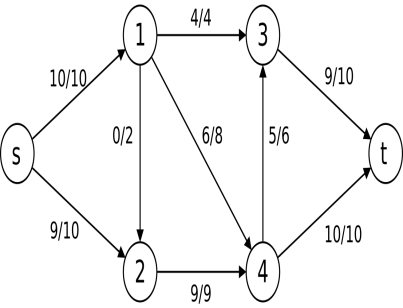
Пример работы алгоритма:

Рассмотрим пример работы на простой транспортной сети, которая имеет 6 вершин и 9 ребер.

Блокирующий поток помечен синим.





На последнем шаге алгоритм заканчивает работу, так как сток помечен меткой ∞. Получившийся максимальный поток равен 19.