Домашнее задание по АиСД 21.10.2016

студента 594 группы Бородина Максима

О. Алгоритм Малхотры - Кумара - Махешвари

Задача A.

Докажем, что из r можно пустить по исходящим рёбрам добавочный поток $\phi(r)$: предположим, что это не так, и в некоторой вершине, этот добавочный поток не сможет "зайти"в вершину v, или выйти из нее. Но тогда $\phi(v) < \phi(r)$ (по определению потенциала). Аналогично доказывается, что в r может "зайти"добавочный поток требуемой величины (для этого можно в уме развернуть все рёбра, и поменять исток и сток местами)

Задача B.

Для каждой вершины сети храним два двусвязных списка рёбер: исходящие и входящие. Это нужно, чтобы за $O(1)$ удалять ненужные рёбра из списка.

Для каждой вершины посчитаем два "потенциала": по входящим в него рёбрам, и по выходящим. Тогда $\phi(v) = min(\phi_1(v), \phi_2(v))$. Это делается за $O(V + E)$.

После этого запускаем цикл от 1 до V: К началу каждой итерации цикла у нас определена вершина r с наименьшим потенциалом. Теперь проталкиваем поток $\phi(r)$ в две стороны: в сторону $target$, и в сторону $source$. Делаем это следующим образом: рассматриваем исходящее ребро. Если его $capacity - flow < push$, насыщаем его, уменьшаем $push$ на $capacity - flow$, и переходим к следующему ребру. После рассмотрения этой вершины идём к следующей. Таким образом, для каждой вершины после такого проталкивания, у нас будет не более, чем одно ненасыщенное ребро, которое изменялось на этой итерации. Все насыщенные рёбра сразу удаляем (они уже больше не могут повлиять на дальнейший ход алгоритма) и пересчитываем потенциал вершины. После чего, если у какой-то вершины были удалены все входящие или исходящие рёбра, удалим и её, так как через неё уже не может быть больше пропущен поток. В конце каждой итерации переберём все вершины и найдём вершину с наименьшим потенциалом. Заметим, что во время каждой итерации точно удаляется вершина с наименьшим потенциалом. Поэтому за V итераций мы действительно посчитаем весь поток данной сети.

Во время каждой итерации алгоритм делает $O(V+T)$ действий, где T - количество удалённых на этой итерации рёбер. Итого, за весь цикл будет выполнено $O(V^2+E)=O(V^2)$ действий

Заметим, что данное решение работает для слоистой сети.

С помощью обхода в ширину для начального графа мы можем построить слоистую сеть за $O(V+E)$ перед каждым запуском алгоритма на такой сети. Всего требуется не более V фаз построения слоистой сети(так как после каждой фазы увеличивается кратчайшее расстояние от $source$ к $target$, а оно не может быть больше, чем V). Итоговая асимптотика: $O(V*(V+E+V^2))$, то есть $O(V^3)$