1 Обязательная часть

1. ***Одиночество в сети.***

   Находим макс поток. Находим мин разрез. Теперь пытаемся выкидывать из него ребра. А точнее берем каждый раз ребро, делаем его пропускную способность $+\inf$. В полученном графе находим опять макс поток. Если новый равен найденному в первый раз, то единственности нет. Если же больше, то продолжаем заменять ребра в первом разрезе. Если по итогу так и не нашли такого же разреза, то единственность выполняется.

2. ***Грузовичкофф.***

   Мы умеем доводить все грузовики до конца за время $K + V$. Находим единичный поток (то есть даем проехать одному грузовику), а потом по нему же с задержкой в одну единицу времени пускаем все остальные. Давайте попробуем проехать быстрее.

   Возьмем наши V вершин, поставим их рядочком и продублируем $K + V$ раз это все дело. Получим $K+V$ слоев. Ребра (единичные!) будем проводить только между соседними слоями, все в соответствии с ребрами изначального графа. Еще добавим бесконечные ребра. Тоже между соседними слоями, но они соединяют копии одной вершины. Это нужно, чтобы позволить грузовичкам не переезжать каждый шаг, а стоять на месте какое-то время. Поставим исток - вершина, соединенная ребром размера K с начальной вершиной в первом слое. В полученном графе $\mathcal{O}((K + V)V)$ вершин и $\mathcal{O}(K + V)E$ ребер.

   Теперь заметим, что за $\mathcal{O}((K + V)K(K + V)E)$ мы можем найти минимальное время, просто пускаем Форда-Фалкерсона, перебирая слой, на котором могут закончить все K грузовиков. Тот слой, начиная с первого, на котором впервые сможем набрать поток величины K и будет ответом. Можем улучшить время работы до $\mathcal{O}(log^{K+V} \cdot K(K + V)E)$. Это получился полином, который хотели в первом пункте. Но хочется еще чуть-чуть улучшить время.

   Заметим, что каждый раз искать новый поток — пустая трата времени. Мы можем старый найденный по бесконечному ребру переместить в новый слой и теперь как бы продолжить искать доп потоки. Как только смогли найти поток K — нашли нужное время. Работает это все дело за $\mathcal{O}((K + V)(K + V)E)$. Но мы же хотим еще быстрее. Мы знаем границу сверху — $K+V$. А может найдем границу снизу? Очевидно, что это самый быстрый путь в изначальном графе от начальной точки к конечной. Этот путь можно найти бфсом, так получили нижнюю границу T. Поймем, что умеем точно ездить за $T + K$. Ну теперь давайте перебирать начинать наши слои не с первого, а с Tго. И максимум на $T + K$ом мы найдем ответ. Тогда это все будет работать за $\mathcal{O}(K(K + V)E)$, чего и добивались.

3. ***Девочки много хотят.***

   Делаем такой графичек: исток, из него единичные ребра во всех мальчиков, из них единичные ребра в соотвествующих девочек (которым они нравятся), из них единичные ребра в нудных собачек, из всех собачек единичные ребра в сток. Теперь возьмем и найдем макс поток.

   Казалось бы, все хорошо. Но сейчас одна девочка может радостно дркжить с двумя мальчиками и собачками одновременно, нечестно как-то. Разрежем каждую девочку пополам, в левую половину приходят ребра от мальчиков, из нее единичное ребро в правую половину, а из нее ребра в собачек. Теперь у каждой девочки может быть не более одного мальчика с собачкой.

4. ***Неорцикл.***

Цикл через две данные вершины в неорграфе... Что же это такое. О, например, два пути в орграфе. Давайте скажем, что одна из вершин — исток, вторая — сток. Для каждого ребра скажем, что его обратное имеет не нулевую пропускную способность, а такую же, как у прямого. Это мы из неор графа сделали ор. Теперь хотим вершинно простой. Ну давайте каждую вершину представим в виде двух (как делали с девочками в предыдущей). У всех ребер пропусная способность один. И последний шаг, найти два пути. Просто найдем поток величины два. Ура, решили задачу.

5. ***Орцикл.***

Давайте сведем NP-полную задачу поиска вершинно непересекающихся путей между двумя парами вершин в ориентированном графе к этой. Пусть хотим найти пути из A в B и из C в D. Давайте добавим к ним две вершинки E и F и проведем ребра из B в E, из E в C и из D в F, из F в A. Что мы теперь хотим? Хотим найти вершинно простой цикл через вершины E и F. Если получится, то удалим все добавленные нами вешины/ребра, получим два искомых вершинно непересекающихся пути.

6. ***Глобальный разрез.***

Что значит, что граф стал несвязным? Значит из какой-то вершины одной части в какую-то вершину другой части нельзя пустить положительный поток (если сделать всех ребер пропускную по 1). Проблема в том, что мы не знаем заранее, что это за вершины, что макс поток из одной в другую первым из всех возможных обозначит несвязность.

Ну а зачем нам знать, времени ведь полином, давайте просто зафиксируем одну вершину-исток (она ведь в одной из частей по итогу точно окажется) перебирать все возможные ее пары стоки. Для каждой пары находим макс поток, из него берем мин разрез любой. Смотрим на количество ребер в этом мин разрезе. Если взять по одному концу каждого ребра и выкинуть его, то получится несвязный граф. Сравниваем все полученные V количеств ребер в разрезах, берем минимальное.

Время работы? V этапов (все пары сток). На каждом делаем поиск потока (Tflow), дфсом ищем мин разрез ($\mathcal{O}(V+E)$), в мин разрезе выбираем по концу ребра ($\mathcal{O}(V)$). Итого это как раз полиномчик.

7. ***Округление матрицы.***

Давайте эксперимента ради округлим все-все-все вниз. Теперь посмотрим на каждую строку и столбец. Для каждого из них запомним такое число delta, что если прибавить его, то будет корректное округление вниз (соответственно delta+1 дает округление вверх). Теперь построим такой волшебный графичек: исток s, из него выходят ребра в вершины, соотвествутющие строкам, из них выходят ребра во вторую долю, отвечающую за столбцы, из "столбцов"вызодят ребра в сток.

Поставим на ребра такие ограничения: по ребрам из стока должен пройти поток размером от соответствующего delta до delta+1, по ребрам из строк в столбцы размером от 0 до 1 (0 символизирует округление вниз, а 1 вверх), по ребрам в сток тоже размером от соответствующей delta до delta+1. Ну все, нашли LR поток, решили задачу.

8. ***Все мы делим пополам.***

   Ну уже очевидно строим двудольный графичек. Ребра между рабочими и работами имеют пропускную способность 1. Добавляем сток, соединяем его с людьми, исток, к нему от работ единичные ребра. Заметим, что от пропускных способностей источных ребер будет зависеть то, сколько работ максимум может сделать рабочий. Хотим это число минимизировать. Ну давайте перебирать.

   Пусть сначала способность 1. Ищем макс поток. Если он равен кол-ву работ, то мы распределили работы по рабочим. Если же он меньше, то увеличиваем способности на 1, пробуем теперь увеличить поток. Делаем так до тех пор, пока поток не станет m. Работает это все дело за $\mathcal{O}(nm^2)$, так как доп путь ищем за nm, а искать придется не больше, чем m раз.

2 Дополнительная часть

1. ***Одиночество в сети. Часть вторая.***

   Как мы обычно ищем мин разрез? Находим макс поток, из истока пускаем по ненасыщенным ребрам дфс. Давайте теперь сделаем второе действие: пустим дфс по ненасыщенным ребрам из стока. Тогда если два разреза, которые мы нашли, не совпадают, то единственность не выполняется (ведь их два). Если же совпадают, то единственно. Почему?

   Доли совпали по двум дфсам, то есть внутри каждой доли можно дойти до вершины только по ненасыщенным ребрам. Заметим, что все ребра из разреза обязательно насыщены. Пусть есть другой мин разрез, тогда хоть одно его насыщенное ребро сейчас лежит внутри одной из долей. Тогда существует путь только по ненасыщенным ребрам от одного конца ребра в другой, понятное дело этот путь хоть раз пересечет наш предполагаемый разрез. Но тогда в разрезе есть ненасыщенное ребро, противоречие. Вывод: разрез единственнен.

   Как максимизировать S? Возмем то S, что получилось при дфсе из стока. Внутри доли T так получившейся все вершины достижимы по ненасыщенным ребрам, то есть ни одну из них нельзя перекинуть в S, то есть T минимально, следовательно, S максимально.