Меня зовут Михаил Енц, я студент 431 группы

Тема моей курсовой работы – сравнение некоторых алгоритмов поиска максимального потока

**Слайд 2. Определения (Ориентированный граф)**

Ориентированный граф – это пара: конечно непустое множество V – вершины графа. На котором задано бинарное отношение, элементы которого называются дугами.

На слайде представлен пример орграфа

**Слайд 2.5 Определения (Взвешенный ориентированный граф)**

Взвешенный ориентированный граф – это орграф, где каждой дуге поставлено некоторое число, называемое весом дуги. Вес мб положительным, нулевым или отрицательным.

**Слайд 3. Определения (Транспортная сеть)**

Транспортной сетью называется ориентированный граф, в котором каждая дуга имеет *неотрицательный* вес*,* называемыйпропускную способность.

Вершина орграфа, недостижимая ни из какой другой вершины, называется истоком.

Вершина орграфа, из которой недостижима ни одна другая вершина, называется стоком.

На слайде представлен пример транспортной сети с одним истоком – 1. И одним стоком – 8.

**Слайд 4. Определения (Поток)**

Потоком в транспортной сети является функция удовлетворяющая трем условиям:

1. Ограничение пропускной способности, означает то, что величина потока проходящая по дуге не может превышать пропускную способность этой дуги.
2. Антисимметричность. Величина потока из вершины в вершину противоположна величине потока из вершины в вершину .
3. Сохранение потока. Для всех узлов, кроме истока и стока. Поток не изменяется при прохождении через вершину, т.е. входящий поток равен выходящему.

**Слайд 6. Определения (Максимальны поток)**

Поток называется максимальным, если он больше или равен любого потока в сети.

Задача о максимальном потоке формируется следующим образом: в заданной сети найти поток максимальной величины.

На слайде представлен пример максимального потока в сети. Максимальны поток равен 36.

**Слайд 7. Алгоритм Форда-Фалкерсона.**

Введем определение. Остаточная сеть – это сеть, в которой каждая дуга допускает увеличение потока. Т.е. для каждой дуги справедливо, что разность пропускной способности и текущего потока (проходящего по этой дуге) положительна.

Путь из вершины в вершину называется увеличивающим путем, если для каждой дуги остаточная пропускная способность положительна. Другими словами, если каждая дуга этого пути принадлежит остаточной сети.

**Слайд 8. Алгоритм Форда-Фалкерсона (Остаточная пропускная способность)**

Максимальная величина, на которую можно увеличить поток вдоль каждой дуги увеличивающего пути , называется остаточной пропускной способностью. И вычисляется, как минимум из всех пропускных способностей дуг принадлежащих пути .

На слайде представлен пример увеличивающего пути в транспортной сети. Величина потока, который можно пустить по этому пути равен 12.

**Слайд 9. Алгоритм Форда-Фалкерсона (Сам алгоритм)**

Алгоритм ФФ решает задачу поиска максимального потока в сети. Он является итеративным. Вначале величине потока присваивается значение 0. На каждой итерации величина потока увеличивается посредством поиска некоторого увиливающего пути и последующего увеличения потока вдоль этого пути.

Как только увеличивающих путей не станет, говорим, что полученный поток является максимальным.

**Слайд 10. Алгоритм Диница.**

Алгоритм Диница является модификацией алгоритма ФФ. Единственное отличие заключается в том, что при поиске увеличивающего пути на шаге 2 в алгоритме ФФ производится поиск кратчайшего пути, например, алгоритмом поиска в ширину.

Время выполнения алгоритма Диница, если применять алгоритм поиска в ширину, составляется , в то время как алгоритм ФФ в общем случае выполняется за время .

**Слайд 11. Алгоритм проталкивания предпотока. (Предпоток)**

Алгоритм проталкивая предпотока также решает задачу поиска максимального потока. Но в отличии от алгоритма ФФ и алгоритма Диница использует понятие предпотока. Алгоритм обрабатывает вершины по одной, рассматривая только соседей данной вершины в остаточной сети.

Предпотоком является некоторая величина, которая может хранить в вершине и при позволяющих обстоятельствах может быть распределена по соседним вершинам. Если предпоток положителен, то вершина считается переполненной.

**Слайд 12. Алгоритм проталкивания предпотока. (Две операции)**

Алгоритм оперирует двумя основными операциями – операция проталкивания предпотока и операция подъема вершины.

**Слайд 13. Алгоритм проталкивания предпотока.**

В транспортной сети функция является функцией высоты, высота истока равно количеству вершин в сети, высота стока равна нулю и выполняется неравенство, что для любой вершины , разность по модулю с высотой ее соседа . Также дуга между этими вершинами должна принадлежать остаточной сети.

**Слайд 14. Алгоритм проталкивания предпотока.**

Операция проталкивания применяется только тогда, когда вершина имеет положительный избыток и остаточная пропускная способность дуги, по которой собираемся протолкнуть поток, строго положительна. И высота вершины на единицу больше высоты вершины . Тогда можно увеличить поток из u в v на величину равную минимуму из избытка вершины и остаточной пропускной способности дуги , при этом избыток не становится отрицательным и не будет превышена пропускная способность дуги.

**Слайд 15. Алгоритм проталкивания предпотока.**

Операция подъема применима только тогда, когда вершина переполнена и все соседние вершины из остаточной сети имеют высоту больше чем вершина .

На слайде представлен второй шаг алгоритма и состояние высот для каждой вершины. Вторая вершина имеет высоту 1, так как на предыдущем шаге вторая вершина имела высоту 0, для того чтобы протолкнуть предпоток, необходимо поднять эту вершину на высоту большую чем минимум из высот её соседей плюс 1

**Слайд 16. Алгоритм проталкивания предпотока. (Инициализация)**

На слайде представлен алгоритм инициализации начального предпотока.

**Слайд 17. Алгоритм проталкивания предпотока. (Сам алгоритм)**

На первом шаге инициализируется начальный предпоток

Затем пытаемся протолкнуть предпоток или поднять вершину

**Слайд 18. Интерфейс. (Чтение из файла)**

Была реализована программа, имеющая графический интерфейс, слева представлены доступные действия, в поле параметры можно задать считывание графа из файла, но таким образом можно считать один определенный граф, поэтому...

**Слайд 18,5. Генератор графов.**

В ходе работы был реализован генератор графов. Генератор имеет параметризацию:

Для генерируемого графа можно задать количество вершин, дуг, можно задать максимальное значение пропускной способности, при этом для каждой дуги будет присвоена пропускная способность в промежутке от 1 до этого максимального значения, также можно задать фиксированную величину пропускной способности.

Гарантируется, что созданный граф представляет собой сеть, где есть один исток – нулевая вершина, один сток – последняя вершина и точно существует путь из истока в сток.

**Слайд 19 Визуализация графов (Диница).**

После считывания из файла или генерации графа можно применить один из реализованных алгоритмов и отобразить граф с найденным на нем максимальным потоком.

Цвет каждой дуги отображает степень насыщенности, чем более светлая дуга, тем больший поток проходит по этой дуге.

На данном слайде найден поток алгоритмом Диница.

**Слайд 20 Визуализация графов (ПП)**

... а на этом алгоритмом проталкивание предпотока. Можно заметить, что найденный поток различается, те он проходит перераспределяет поток по дугам с разной интенсивностью, но ответ от этого не меняется.

**Слайд 21 Визуализация графов (Еще пример)**

**Слайд 22 Визуализация графов (Только путь)**

Также можно отобразить только те дуги, по которым проходит ненулевой поток

**Слайд 250. Сравнение и анализ.**

В ходе работы была проведена серия экспериментов, замеряя время работы на одинаковых сетях для обоих алгоритмов.

На слайде приведена часть таблицы экспериментов. Параметры на слайде: вершины, дуги, фиксированная пропускная способность, максимальная пропускная способность, время Диница, время проталкивания предпотока.

Случайным образом создавался граф, с определенными параметрами и считалось время. Учитывались плотные и разреженные графы.

По итогу можно заметить то, что алгоритм Диница работает быстрее алгоритма проталкивания предпотока, несмотря на то, что они имеют одинаковую асимптотику. Так как проталкивание предпотока имеет большую скрытую константу в своей асимптотике.

И так как асимптотики одинаковые, применяя эти алгоритмы на практике, можно выбрать заведомо более медленный алгоритм, что нежелательно)