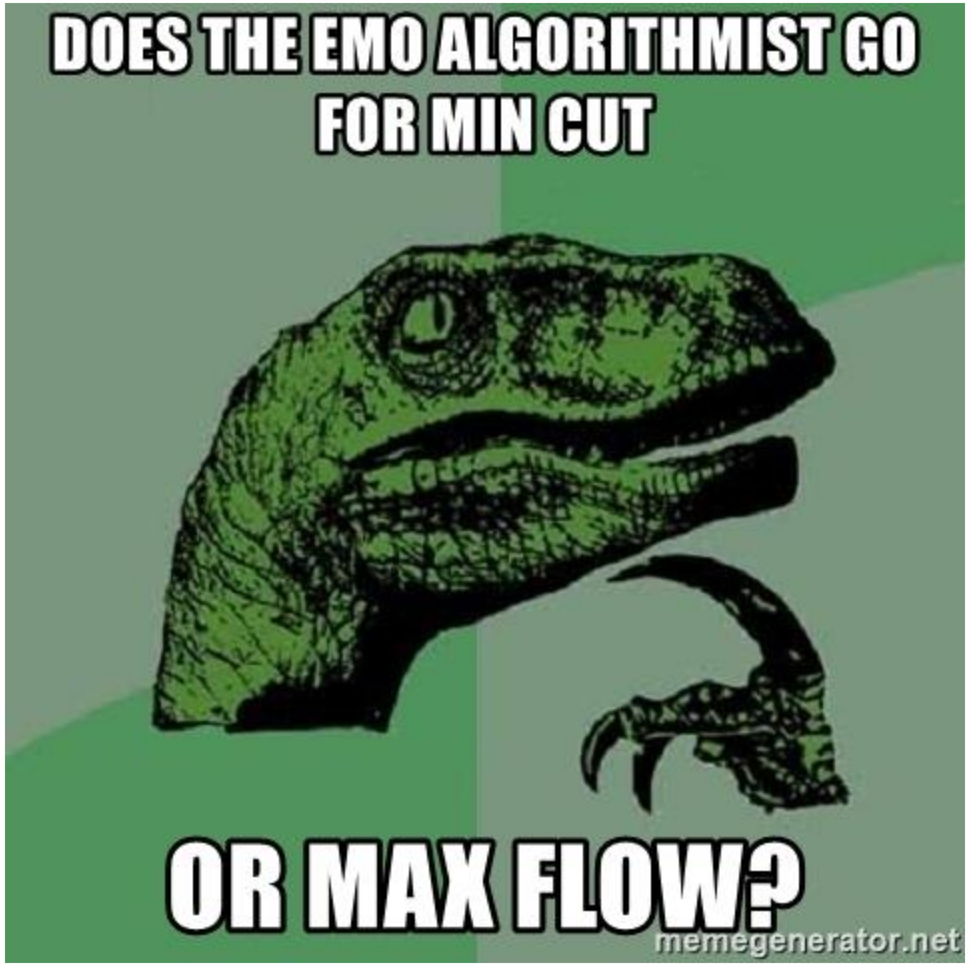
академия больших данных

mail.ru group

Графы – 4. Введение в потоки.

Шовкопляс Григорий

Введение в алгоритмы и структуры данных



Задача о поиске максимального потока

Основные определения

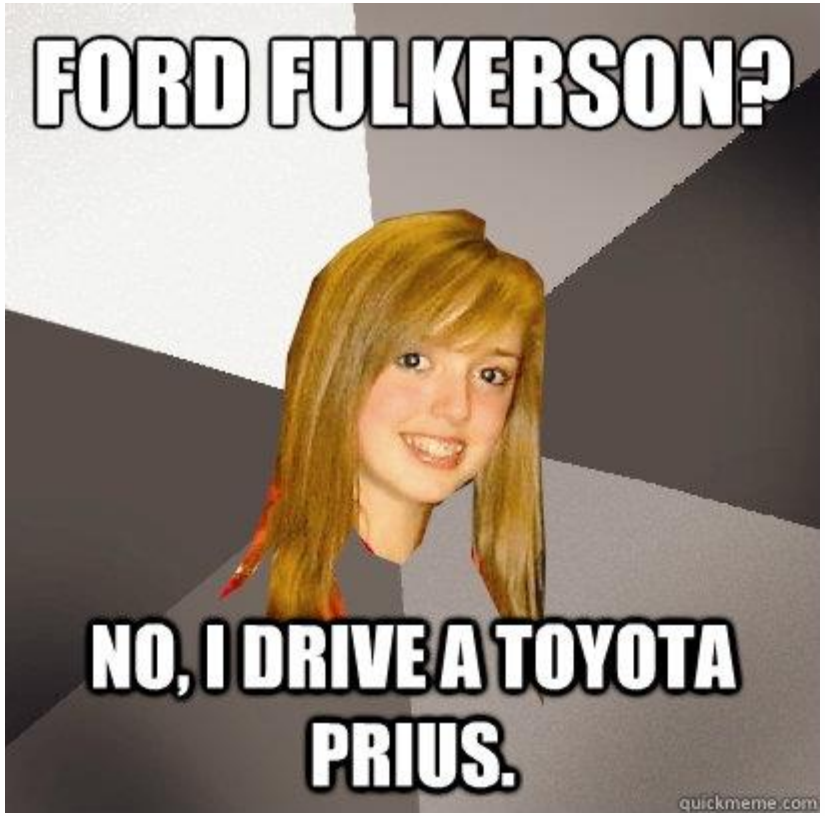
- Дан ориентированный граф $G(V, E)$
- s – «исток»
- t – «сток»
- $c(u, v)$ – пропускная способность ребра из u в v
- Если ребра нет $c(u, v) = 0$
- Пусть нет петель и кратных ребер
- Четверка (G, c, s, t) – сеть
- Что же такое поток?

Поток

- Поток: функция $f: V \times V \to R$
- Свойства:
  - Непревышение пропускных способностей: $f(u, v) \leq c(u, v)$
  - Антисимметричность: $f(u, v) = -f(v, u)$
  - «Закон сохранения жижи»: $\sum f(u, \_) = 0$, для всех u, кроме s и t
- $\sum f(s, \_) = \sum f(\_, t) = |f|$ – «величина потока»
- На всякий случай уточним $|f| \geq 0$
- Научимся искать $\max|f|$

Минимальный разрез

- Разрез разбиение V на два множества S и T
- V = S + T
- Рассмотрим такой разрез, что:
    - $s \in S$
    - $t \in T$
- Стоимость разреза $c(S,T) = \sum_{u \in S, v \in T} c(u,v)$
- Заодно научимся искать $\min(c(S,T))$



Алгоритм Форда-Фалкерсона

Алгоритм Форда-Фалкерсона

- Новое определение: *остаточная сеть*
- $C_f = (G', c', s, t)$
- $G' = (V, E')$
- $c'(u, v) = c(u, v) - f(u, v)$
- Если $c'(u, v) = 0$, назовем его насыщенным и его нет в E'
- При этом добавляются новые ребра, где $c'(u, v) > 0$

Теорема Форда-Фалкерсона

- Следующие три утверждения эквивалентны:
  - f – максимальный поток в сети
  - В остаточной сети C_f нет пути $s \to t$
  - Существует разрез (S, T), что $|f| = c(S, T)$

Теорема Форда-Фалкерсона (1 ⇨ 2)

- Докажем от противного
- Пусть в остаточной сети существует путь $s \to t$
- Тогда его можно добавить к потоку
- Все свойства сохранятся, величина больше

Теорема Форда-Фалкерсона (2 ⇨ 3)

- В остаточной сети нет пути
- Обозначим $S = \{u \mid \text{в остаточной сети есть путь } s \to u\}$
- $T = V \setminus S$
- Оба множества непустые, s лежит в S, t лежит в T
- Ребра, пересекающие разрез, насыщены
- $c(S,T) = |f|$

Теорема Форда-Фалкерсона (3 ⇨ 1)

- Любой разрез $\geq$ любого потока
- Но мы имеем пару из потока и разреза, что $|f| = c(S,T)$
- Значит, поток максимален, а разрез минимален

Алгоритм Форда-Фалкерсона

- $f_0 = 0$, начальный поток, будем его увеличивать
- i-й шаг цикла:
- Случай 1:
  - Если в сети $C_{f_{i-1}}$ существует дополняющий путь из s в t, найдем при помощи обхода в глубину
  - f' – минимальная пропускная способность на этом пути
  - Тогда $f_i = f_{i-1} + f'$, а также на всех ребрах пути обновим $f(u, v)$, чтобы неявно получить C_{f_i}
- Случай 2:
  - Иначе по теореме нашли максимальный поток

Алгоритм Форда-Фалкерсона

- Оценим время работы
- Недетерменированно ищем дополняющий путь
- Проиграем на «перечеркнутом ромбике»
- Время работы будет $O(|f| \times (V + E))$

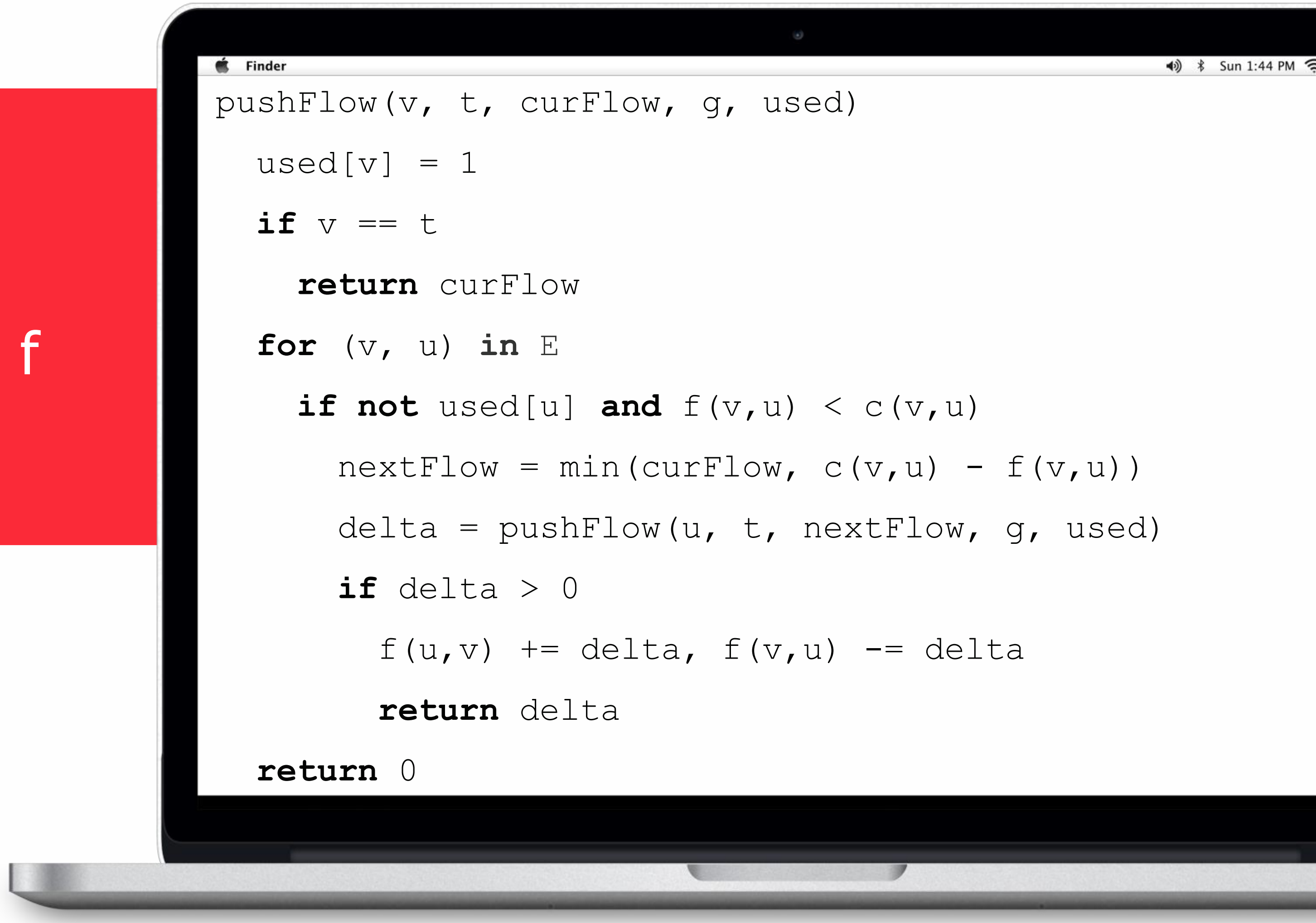
Алгоритм Форда-Фалкерсона

- Пока есть путь, пушим поток

```
while true
  used.assign(n, 0)
  delta = pushFlow(s, t, INF, g, used)
  if delta > 0
    ans += delta
  else
    break
```

Алгоритм Форда-Фалкерсона

- pushFlow по факту обычный DFS с обновлением значений f



Алгоритм Форда-Фалкерсона

- Как найти минимальный разрез?
- $S = \{u \mid \text{в остаточной сети есть путь } s \to u\}$
- $T = V \setminus S$

- Еще довольно популярная задача, поиск числа непересекающихся путей из s в t

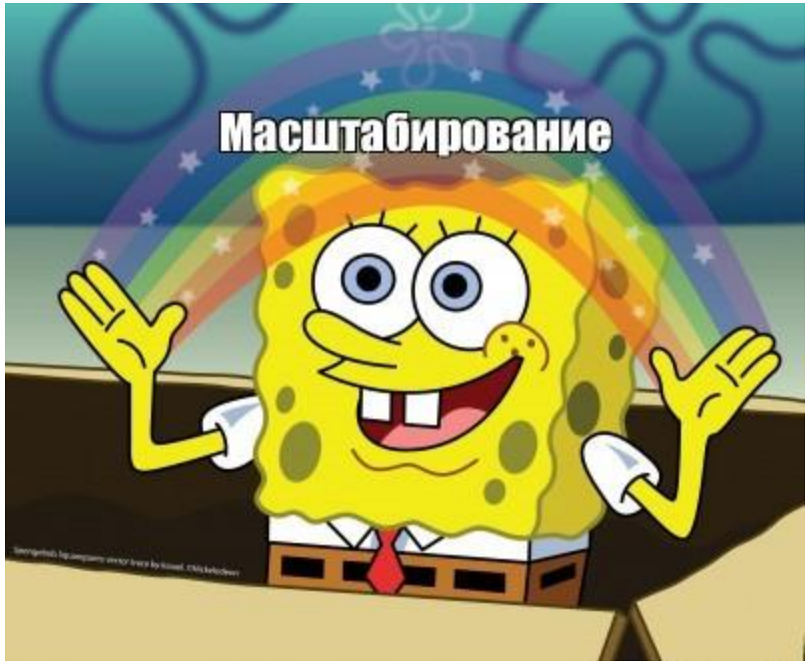
Алгоритм Эдмондса-Карпа

Алгоритм Эдмондса-Карпа

- Раньше дополняющий путь искали обходом в глубину
- А что если использовать поиск в ширину?
- На самом деле теперь мы будем выбирать дополняющий путь минимальный по количеству ребер и это приведет к успеху

Алгоритм Эдмондса-Карпа

- Назовем ребро «критическим», если оно полностью насытилось в результате добавления пути
- Каждое ребро будем критическим не больше, чем V раз
    - Расстояние до вершины не уменьшается
    - Расстояние до вершины не может быть больше $|V| - 1$
- Следовательно, найдем не больше VE путей
- Время работы $O(VE \times (V + E)) = O(VE^2)$



Масштабирование потока

Масштабирование потока

- Лайфхак, который можно применить к любому алгоритму поиска потока, даже к тем, что мы не проходили!
- Пусть $U = \max(c(u, v))$
- Сделаем ограничение на проталкиваемый поток снизу Δ

Масштабирование потока

- Будем использовать только ребра $c(u, v) \geq \Delta$
- $\Delta = \{2^{\log U}, 2^{\log U - 1}, \dots, 4, 2, 1\}$
- $\max|f| \leq |f_k| + 2^k E$, где $|f_k|$ – поток при масштабе $\Delta = 2^k$
  - Все ребра через минимальный разрез в остаточной сети $c'(u, v) \leq \ 2^k$, и суммарно их не больше E
- Суммарное количество увеличивающих путей $O(E \log U\ )$
  - Дополняющий путь не меньше Δ
  - Значит дополняющих путей не больше $2E$
- Таким образом сложность можно сократить до $O(E^2 \log U)$

Все!