Борувка обычный и случайный

Денис Осипов

Не очень официальная версия 2 для зума, не для ВКР

18 сентября 2020

Постановка задачи

Дан связный граф $G = (V, E)$. У каждого ребра (u, v) есть неотрицательный вес $w(u, v)$.

Определение

Остовное дерево – подмножество ребер, покрывающее все вершины графа
Минимальное остовное дерево – остовное дерево минимального суммарного веса.

Обозначение: **MST** – minimum spanning tree.

Задача: найти минимальное остовное дерево.

1 Постановка задачи

2 Обычный алгоритм Борувки

- Описание алгоритма: шаг Борувки
- Безопасные ребра
- Лемма о безопасных ребрах
- Теорема о корректности
- Время работы

3 Линейная случайная модификация

- F-тяжелые ребра
- Лемма о тяжелых ребрах
- Матожидание числа F-легких ребер в случайном графе
- Описание алгоритма
- Корректность
- Время работы

Шаг Борувки

Шаг Борувки – алгоритм сведения $MST(V,E) \to MST(\leq \frac{V}{2}, \leq E)$.

Сколько раз его надо применить, чтобы построить миностов? $O(\log V)$ раз.

Алгоритм Борувки = применить шаг Борувки $O(\log V)$.

Перед первым шагом Борувки как-то **пронумеруем** ребра.

Шаг Борувки

1. Для каждой $v \in V$ помечаем смежное ребро минимального веса. Если их **несколько**, берем ребро с наименьшим номером.
2. Определим компоненты связности на графе из помеченных ребер (например, поиском в глубину)
3. Каждую КС стянем в одну вершину. Некоторые ребра при этом станут петлями или мультиребрами.
4. Все петли уберем, а в мультиребрах оставим только ребра минимального веса.

1. Постановка задачи
2. Обычный алгоритм Борувки
   - Описание алгоритма: шаг Борувки
   - Безопасные ребра
   - Лемма о безопасных ребрах
   - Теорема о корректности
   - Время работы
3. Линейная случайная модификация
   - F-тяжелые ребра
   - Лемма о тяжелых ребрах
   - Матожидание числа F-легких ребер в случайном графе
   - Описание алгоритма
   - Корректность
   - Время работы

Безопасные ребра

Определение

Пусть $G' \subseteq G$ – подграф. Ребро $e \notin G'$ *безопасное*, если $G' \cup e$ содержится в каком-нибудь минимальном остовном дереве.

Т.е. «безопасное» – то, которое можно взять в миностов.

Для каждого из трех алгоритмов MST (Прима, Крускала, Борувки) есть свое достаточное условие на безопасность ребра.

Это условие лежит в основе этих алгоритмов, которые по существу являются «жадниками» относительно этих условий.

1. Постановка задачи
2. Обычный алгоритм Борувки
   - Описание алгоритма: шаг Борувки
   - Безопасные ребра
   - Лемма о безопасных ребрах
   - Теорема о корректности
   - Время работы
3. Линейная случайная модификация
   - F-тяжелые ребра
   - Лемма о тяжелых ребрах
   - Матожидание числа F-легких ребер в случайном графе
   - Описание алгоритма
   - Корректность
   - Время работы

Какие ребра берет алгоритм Борувки?

Для алгоритма Борувки это условие выглядит так.

Лемма (о безопасных ребрах алгоритма Борувки)

Для всякой вершины $v \in V$ и любого минимального остовного дерева T хотя бы одно смежное с v ребро минимального веса входит в T.

Доказательство. НУО хотя бы одно ребро не минимальное (иначе доказывать нечего).

Пусть T – какой-то миностов, не содержащий ни одного минимального ребра. Мы хотим перестроить его в более легкий остов и получить противоречие с минимальностью. Пусть $(v, w) \in T$ – не минимальное ребро, $(v, u) \notin T$ – минимальное ребро. T – остов, значит $\exists!$ путь P: $u \to w$.

Если $P \not\ni (v, w)$, то уберем из T ребро (v, w) и добавим (v, u). Связность T сохранилась: вместо ребра (v, w) есть путь $(v, u) + P$.

Если $P \ni (v, w)$, то рассмотрим другое ребро $(v, w_1) \in P$. По предположению оно не минимальное. Уберем его из T и возьмем (v, u). Связность T снова сохранилась, так как есть путь $v \to w_1$: $(v, u) + (P \setminus ((w_1, v) + (v, w))$. □

Теорема о корректности

Пометим для каждой вершины ребро мин.веса, мин.номера →
Стянем каждую КС на помеченных → Уберём петли и сократим кратные ребра

Теорема о корректности

Пусть шаг Борувки из графа G получил граф G'. T' – миностов G', S – множество ребер, помеченных шагом Борувки. Тогда $S \cup T'$ – миностов G.

Доказательство. Выведем теорему из леммы (потом докажем):

Лемма

Множество помеченных ребер S образует лес в G.

Доказательство теоремы. Каждому КС-дереву леса S соответствует вершина в графе G'. T' соединяет все вершины G', то есть все КС-деревья графа на помеченных ребрах G. Значит, $T' \cup S$ дерево.

С другой стороны, по построению $T' \cup S$ подмножество какого-то миностова. ☐

Пометим для каждой вершины ребро мин.веса, мин.номера $\to$ Стянем каждую КС на помеченных $\to$ Уберём петли и сократим кратные ребра

Лемма

Множество помеченных ребер S образует лес в G.

Доказательство. Пусть есть цикл. Ориентируем: если v «выбрала» (v, u), то $v \to u$.

Получился орцикл. Действительно, $out(v) = 1$ (так «выбирали»), $in(v) + out(v) = 2$ (цикл). Значит $in(v) = 1$.

Имеем орцикл $v_1 \to v_2 \to \ldots \to v_k \to v_1$. По алгоритму «выбора» $w(v_j, v_{j+1}) \le w(v_{j-1}, v_j)$ для всех $j = 1..k$ (считаем $v_{k+1} \equiv v_1$). Отсюда все w равны.

По смыслу алгоритма при равных весах выбирается меньший номер. Имеем $\#(v_j, v_{j+1}) < \#(v_{j-1}, v_j)$ для всех $j = 1..k$, но это уже невозможно. □

Мы доказали корректность алгоритма Борувки. Осталось оценить время работы.

1. Постановка задачи
2. Обычный алгоритм Борувки
    - Описание алгоритма: шаг Борувки
    - Безопасные ребра
    - Лемма о безопасных ребрах
    - Теорема о корректности
    - Время работы
3. Линейная случайная модификация
    - F-тяжелые ребра
    - Лемма о тяжелых ребрах
    - Матожидание числа F-легких ребер в случайном графе
    - Описание алгоритма
    - Корректность
    - Время работы

Время работы шага Борувки

1. Для каждой $v \in V$ помечаем смежное ребро минимального веса. Если их **несколько**, берем ребро с наименьшим номером.
2. Определим компоненты связности на графе из помеченных ребер (например, поиском в глубину)
3. Каждую КС стянем в одну вершину. Некоторые ребра при этом станут петлями или мультиребрами.
4. Все петли уберем, а в мультиребрах оставим только ребра минимального веса.

По шагам:

1. Однократный просмотр всех смежных ребер у всех вершин: $O(E+V)$
2. Поиск в глубину: $O(E+V)$
3. Переназначение вершин и переподключение ребер: $O(E+V)$
4. Просмотр всех ребер: $O(E)$

Всего $O(E+V)$. В связном графе $V \leq E+1$, так что можно и $O(E)$.

Весь алгоритм: $O(E \log V)$

Перерыв?

1. Постановка задачи
2. Обычный алгоритм Борувки
    - Описание алгоритма: шаг Борувки
    - Безопасные ребра
    - Лемма о безопасных ребрах
    - Теорема о корректности
    - Время работы
3. Линейная случайная модификация
    - F-тяжелые ребра
    - Лемма о тяжелых ребрах
    - Матожидание числа F-легких ребер в случайном графе
    - Описание алгоритма
    - Корректность
    - Время работы

Подготовка к линейному алгоритму

Мы улучшим этот алгоритм, и он будет работать за $O(E)$ в среднем.

Пусть F – лес в G.

Определение

Ребро (u, v) называется F-**тяжелым**, если $w_F(u, v) < w(u, v)$,
где $w_F(u, v)$ – максимальное ребро на единственном пути $u \to v$ по F.
Если такого пути нет, то $w_F(u, v) = \infty$.

Примеры

- Если $e \in F$, то e – F-легкое.
- Ребро e F-тяжелое $\iff$ существует путь $u \to v$ в F, и все ребра этого пути строго легче ребра (u,v).

Основная идея:

Лемма о тяжелых ребрах

Пусть F – **любой** лес в G. Если ребро F-тяжелое, то оно не лежит в миностове G.

Лемма о тяжелых ребрах

Пусть F – **любой** лес в G. Если ребро F-тяжелое ($w_F(u, v) < w(u, v)$), то оно не лежит в миностове G.

Доказательство. Фиксируем лес F. Пусть T – миностов, $T \ni (u, v)$ – F-тяжелое ребро. Мы хотим перестроить миностов в более легкий остов и получить противоречие.

Удалим из дерева T ребро (u, v) и тогда дерево распадется на две КС. С другой стороны, по определению F-тяжелого ребра, в F есть путь P: $u \to v$, все ребра которого строго легче, чем (u, v).

Этот путь перейдет из одной КС в другую, значит в нем найдется одно ребро e, соединяющее две компоненты связности.

Таким образом, в T достаточно заменить (u, v) на e, чтобы получить более легкий остов. ☐

Еще одна страшная теорема

Пусть G – граф, $0 < p < 1$

Определение

Случайный граф $G(p)$ строится как граф на тех же вершинах так: каждое ребро G попадает в $G(p)$ с вероятностью p. Других ребер нет.

Замечание: связность $G(p)$ никак не гарантируется.

Теорема

Существует остовный лес F графа $G(p)$, при котором матожидание числа F-легких ребер в G не превосходит n/p.

Доказательство-1: подготовка

Упорядочим все ребра в порядке неубывания весов: $e_1, \ldots, e_m$.

Граф $G(p)$ будем строить, рассматривая и принимая/отвергая ребра именно в этом порядке.

Минsocтов F строим параллельно с $G(p)$. Изначально F это безреберный граф на вершинах G. Если e_i взято в $G(p)$ и на тот момент соединяло две компоненты связности F, то берем e_i в F.

Ребро $e_i = (u, v)$ F-легкое для G в момент рассмотрения $\iff$
В момент рассмотрения e_i в лесе F нет пути $u \to v$ $\iff$
Оно соединяет две компоненты связности F $\iff$
Если оно попало в $G(p)$, то оно взято в F.

С другой стороны,

Ребро e_i F-легкое для G в момент рассмотрения $\iff$
Ребро e_i F-легкое для G в конце построения $G(p)$.

Всякое F-легкое ребро либо не попало в $G(p)$, либо попало и в $G(p)$, и в F.

Доказательство-2: случайный процесс

Всякое F-легкое ребро либо не попало в $G(p)$, либо попало и в $G(p)$, и в F.

Определим k-тую фазу построения: всё то время, когда в F ровно $k-1$ ребро.

Всякое F-легкое ребро с вероятностью p попадает в $G(p)$, а значит и в F.

Сколько рассмотрим F-легких ребер за k-тую фазу? Сначала какое-то количество F-легких ребер не попадут в $G(p)$, потом одно попадет в $G(p)$ и F – на этом фаза k сразу закончится.

Определим случайную величину A_k – число рассмотренных F-легких ребер за фазу k. Из сказанного выше она имеет геометрическое распределение: $\mathbb{P}(A_k = l) = p(1-p)^{l-1}$.

Стало быть, $\mathbb{E}A_k = \frac{1}{p}$.

Ясно, что число F-легких ребер есть $\sum\limits_{k=1}^{s} A_k$, где $s < n$ – число ребер в итоговом остовном лесе F.

Имеем $\mathbb{E}\sum\limits_{k=1}^{s} A_k = \frac{s}{p} \leq \frac{n}{p}$. □

1. Постановка задачи
2. Обычный алгоритм Борувки
   - Описание алгоритма: шаг Борувки
   - Безопасные ребра
   - Лемма о безопасных ребрах
   - Теорема о корректности
   - Время работы
3. Линейная случайная модификация
   - F-тяжелые ребра
   - Лемма о тяжелых ребрах
   - Матожидание числа F-легких ребер в случайном графе
   - Описание алгоритма
   - Корректность
   - Время работы

Сам алгоритм

Задача: для графа G_1 с n вершинами и m ребрами построить минимальный остовный лес (граф может быть не связен).

Линейный случайный Борувка

1. Применим к G_1 три *шага Борувки*. Получится граф G_2 с $\leq \frac{n}{8}$ вершинами и $\leq m$ ребрами, а также множество S помеченных ребер.
2. Построим случайный граф $G_2\left(\frac{1}{2}\right)$. В нем будет $\leq \frac{n}{8}$ вершин, матожидание числа ребер – $\leq \frac{m}{2}$.
3. Рекурсивно построим миностов F графа $G_2(\frac{1}{2})$.
4. Найдем все F-тяжелые ребра графа G_2. Выкинем их из G_2, получится граф G_3. Матожидание числа ребер G_3 не превосходит $\frac{n}{4}$.
5. Рекурсивно построим миностов F_3 графа G_3. Выдадим ответ: миностов есть $S \cup F_3$.

Корректность

1. Применим к G_1 три *шага Борувки*. Получится граф G_2 с $\leq \frac{n}{8}$ вершинами и $\leq m$ ребрами, а также множество S помеченных ребер.
2. Построим случайный граф $G_2\left(\frac{1}{2}\right)$. В нем будет $\leq \frac{n}{8}$ вершин, матожидание числа ребер – $\leq \frac{m}{2}$.
3. Рекурсивно построим миностов F графа $G_2(\frac{1}{2})$.
4. Найдем все F-тяжелые ребра графа G_2. Выкинем их из G_2, получится граф G_3. Матожидание числа ребер G_3 не превосходит $\frac{n}{4}$.
5. Рекурсивно построим миностов F_3 графа G_3. Выдадим ответ: миностов есть $S \cup F_3$.

Этот алгоритм корректен.

- Шаги 2-3 это просто построение «хорошего» леса, относительно которого мало F-легких ребер.
- На шаге 4 из G_2 выкидываются F-тяжелые ребра (так можно! они не войдут в остов по лемме о тяжелых ребрах)

Таким образом, этот алгоритм – просто более умный алгоритм Борувки.

Время работы

1. Применим к G_1 три *шага Борувки*. Получится граф G_2 с $\leq \frac{n}{8}$ вершинами и $\leq m$ ребрами, а также множество S помеченных ребер.
2. Построим случайный граф $G_2\left(\frac{1}{2}\right)$. В нем будет $\leq \frac{n}{8}$ вершин, матожидание числа ребер – $\leq \frac{m}{2}$.
3. Рекурсивно построим миностов F графа $G_2(\frac{1}{2})$.
4. Найдем все F-тяжелые ребра графа G_2. Выкинем их из G_2, получится граф G_3. Матожидание числа ребер G_3 не превосходит $\frac{n}{4}$.
5. Рекурсивно построим миностов F_3 графа G_3. Выдадим ответ: миностов есть $S \cup F_3$.

По шагам:

1\. $O(n+m)$ 2. $O(n+m)$ 3. $T(\frac{n}{8}, \frac{m}{2})$ 4. ????? 4. $O(n+m)$ 5. $T(\frac{n}{8}, \frac{n}{4})$

Здесь **поверим** в то, что шаг 4 как-то реализуется за $O(n+m)$.

$$T(n,m) = T(\frac{n}{8}, \frac{m}{2}) + T(\frac{n}{8}, \frac{n}{4}) + O(n+m)$$

Простая индукция: $T(n,m) = O(n+m)$.


ВСЁ!