SPb HSE, ПАДИИ, 1 курс, осень 2024/25
Практика по алгоритмам #15

MST и DSU

16 января

Собрано 20 января 2025 г. в 12:21

Содержание

MST и DSU

1. **Дерево кратчайших путей**

   Дан орграф без отрицательных циклов (отрицательные ребра могут быть) и расстояния от s до всех вершин в графе. Восстановить дерево кратчайших путей за $\mathcal{O}(E)$.

2. **Единственность MST**

   Доказать, что в графе с различными весами рёбер MST единственно.

3. **Перестроение MST**

   Дан взвешенный граф G. Дано минимальное остовное дерево на нем.
   У ребра e поменяли вес. Найти новое минимальное остовное дерево за $\mathcal{O}(V + E)$.

   Разберите все 4 *случая!*

4. **Online двудольность**

   Дан неорграф. В него в online добавляются рёбра.
   После каждого добавления говорить, двудолен ли граф?

   a) $\mathcal{O}(m \log n)$ через перекрашивание компонент.
   b) Быстрее через DSU.

5. **Чётность**

   В каждой клетке прямой записано число 0 или 1. Поступает информация: чётность количества единиц на отрезке $[L_i, R_i]$, найти первый запрос, после которого данные противоречивы.

6. **Казалось бы, причём здесь СНМ?**

   У нас есть массив длины n, мы хотим выполнить m запросов вида «покрасить отрезок $[l_i, r_i]$ массива в цвет c_i» и вывести, что получилось в конце. Решите за $\mathcal{O}(n \log m)$.

   Подсказка: попытайтесь выполнять запросы, начиная с последнего.

7. **Второе по весу ST**

   Найти второе по весу остовное дерево: (a) за полином, (b) за $\mathcal{O}(V^2 + E)$.

8. **(*) Дейкстра вышел погулять**

   Дан взвешенный орграф ($n, m \leqslant 10^5, w_e \geqslant 0$). Если мы стартуем в s, идём по некоторому пути и останавливаемся в вершине v, мы получим удовольствие $a_v - w(path(s, v))$. Для каждой вершины графа скажите, куда идти, чтобы получить максимальное удовольствие от прогулки?

9. **(*) Краскал наоборот**

   Пусть дан связный взвешенный неорграф, будем рассматривать его ребра в порядке невозрастания веса и удалять текущее ребро, если связность графа при этом не нарушается. Докажите, что этот алгоритм находит минимальный остов, или придумайте контрпример.

10. **(*) Борувка и MST**

    Верно, ли, что для каждой вершины нужно добавить в MST минимальное по весу смежное ребро? Как на этой идее построить корректный алгоритм для MST?

Разбор задач практики

1. **Дерево кратчайших путей**

   Остовное дерево dfs по ребрам $e = (v, u)$, где $d[v] + w_e = d[u]$.

2. **Единственность MST**

   a) Как в лемме о разрезе: рассмотрим Краскала, пусть в какой-то момент он не берёт минимальное ребро, увидим цикл, сделаем замену, благодаря тому, что веса различны, вес остова строго уменьшится.

   b) **Простой способ.** Строим как-нибудь MST, обозначим его T.

   Строим второе дерево: запустим Краскала, который пытается при равенстве весов сперва брать рёбра не из T. Если новое дерево равно T, то MST единственно.

   Второй способ.

   Запускаем алгоритм Краскала. Внутри обрабатываем ребра группами одинакового веса. Посмотрим на все рёбра веса w. Выберем из них те, которые сейчас соединяют разные компоненты (образованные меньшими ребрами), но в дерево пока не добавим.
   По выбранным рёбрам делаем второй проход, но уже объединяя компоненты и добавляя рёбра в остовное дерево.
   Если какое-то помеченное ребро e_i не добавилось в дерево, значит, его добавление создаёт цикл по рёбрам веса $w \Rightarrow$ MST не единственно. Если ни разу не возникла такая ситуация, MST единственно. Доказательство аналогично (a) «пусть не взяли, сделаем замену...».

3. **Перестроение MST**

   Обозначим наше минимальное дерево T, а меняющееся ребро e.

   - Если **уменьшилось** $e \in T$, или **увеличилось ребро** $e \notin T$, MST не изменилось.

   - Пусть **увеличилось** $e \in T$. Удалим e, получим разрез.
   Добавим минимальное ребро через разрез (им может оказаться снова e).
   Время $\mathcal{O}(E)$ (раскрасить компоненты, перебрать рёбра).
   Корректность. Выбранное нами ребро точно нужно брать по лемме о разрезе, возьмём. Далее запустим Краскала, он возьмёт ровно те же рёбра, что и раньше.

   - Пусть **уменьшилось** $e \notin T$. Рассмотрим $T+e$, e образует цикл. Если в цикле максимальное ребро f больше e, заменим f на e. Время работы = времени поиска максимума на пути.
   Корректность. Если $w_f \leqslant w_e$, шаги Краскала не изменятся. Если $w_f > w_e$, удалим f, получим разрез, e – минимальное в разрезе (если не e и не f, T было не минимальным) $\Rightarrow$ по лемме о разрезе e выгодно взять. Далее Краскал возьмёт ровно те же рёбра, что и раньше.

4. **Online двудольность**

   b) $\mathcal{O}((m + n)\alpha(n))$.

   Способ 1 (основной). У каждой компоненты связности есть одна или две доли. Храним доли в DSU. Если у компоненты две доли, их корни a_i, b_i, то будем хранить «ссылку на вторую долю» `link[a`$_i$`] = b`$_i$`, link[b`$_i$`] = a`$_i$. При добавлении ребра (x, y), три случая:

1. `root(a) == root(b)` ⇒ нечётный цикл.
2. `link[root(a)] == root(b)` ⇒ ребро между долями, ничего не делаем.
3. Ребро между компонентами.
   У нас есть 4 доли – корни и их `link`-и, вызовем два `join`-а, обновим `link`-и.

Способ 2. Храним цвета вершин и компоненты связности в «DSU на деревьях», перекрашиваем лениво. На рёбрах DSU указаны цвета 0 и 1.
Цвет вершины v – `XOR` значений на пути от v до корня DSU.
При `join` ставим на новое ребро 1, если нужно перекрашивание.
При сжатии путей на новое ребро ставим цвет, равный цвету вершины.

5. **Чётность**

Число единиц на отрезке $[L, R]$ чётно, то число единиц на префиксах $[1, L-1]$ и $[1, R]$ имеет одинаковую чётность, иначе разную
⇒ видим граф, где вершины – префиксы, рёбра – отрезки
⇒ задача аналогична предыдущей, только здесь нам сообщают, либо что концы ребра должны иметь одинаковый цвет (лежать в одной доли), либо разный (лежать в разных долях). Каждая компонента связности двудольна, можем, как и в решении выше хранить её, как «пара двух долей».

Решение без двудольности. Будем проводить рёбра с весами 01. СНМ. Каждая компонента — множество в СНМ. Когда приходит ребро $a \xrightarrow{w} b$, если a и b уже в одной компоненте, проверяем, что $w_e =$ `xor` на пути $a \rightsquigarrow b$, а это ровно $up(a) \oplus up(b)$, где $up(a)$ – `xor` на пути от a до корня. $up(a)$ считается СНМ-ом также, как $get(a)$. Когда делаем `join`, проводим ребро не между a и b, а между $get(a)$ и $get(b)$ веса $up(a) \oplus up(b) \oplus w$.

P.S. При больших координатах в offline можно сжать координаты, в online использовать хеш-таблицу (если `p[x]` еще не был определен, то `p[x] = x`).

6. **Казалось бы, причём здесь СНМ?**

Обрабатывая запросы с конца, идём слева направо и красим ещё непокрашенные клетки.
Как быстро пропускать непокрашенные?
Сжатие путей: изначально `p[i]=i+1`, после покраски слева направо путь сжимается.
Сжатие путей, как и СНМ с одной эвристикой даст суммарное время $\mathcal{O}((m+n)\log n)$.

Полноценный СНМ. В СНМ храним области «что-то покрашенное и, возможно, крайняя правая не покрашенная». Для корня множества помним крайнюю правую клетку. Тогда

```
for (int l = right[get(l)]; l <= r; l = right[get(l)]):
    if (color[l] == -1) color[l] = c;
    join(l, l+1);
```

7. **Второе по весу ST**

Обозначим первое по весу T_1, второе по весу T_2.
Переберём (угадаем) $e \in T_2 \setminus T_1$, рассмотрим $e + T_1$, $e(a, b)$ образует цикл.
Найдём f максимальное ребро на пути $a \rightsquigarrow b$ в T_1.
При замене $f \to e$, вес остова меняется на $w_e - w_{f(e)}$, $T_2 = T_1 + e - f$, где $e\colon w_e - w_{f(e)} = \min$.

Корректность. Докажем, что $\forall e$ среди всех остовов, содержащих e, мы нашли минимальный. Мысленно запустим Краскала, он взял ровно те же рёбра, что и при построении T_1. Кроме последнего рассмотренного ребра на цикле – f.

Время работы.
Искать максимальный вес на пути можно за $\mathcal{O}(n)$, тогда получим время $\mathcal{O}(mn)$.
Можно за $\mathcal{O}(n^2)$ предподсчитать все n^2 максимумов, тогда время $\mathcal{O}(n^2 + m)$.
Можно искать максимум на пути ещё быстрее. Даже за $\mathcal{O}(1)$.

8. **(*) Дейкстра вышел погулять**

   Можно как будто для каждой вершины запустить Дейкстру, но это долго.
   Можно пустить одну Дейкстру «с конца»: изначально $d_v = -a_v$, кидаем всех в кучу, и запускаем Дейкстру по обратным рёбрам.

9. **(*) Краскал наоборот**

   Работает. По индукции. Потому что Краскал не возьмёт последнее ребро. Если все веса различны – очевидно. Если бывают одинаковые – сделаем их различными, заменив на $\langle w_e, e \rangle$.

10. **(*) Борувка и MST**

    См. конспект ПМИ.

Домашнее задание

1. **(1) Ремонт дорог**

   Дана страна, состоящая из городов и двусторонних дорог. Каждая дорога или в рабочем состоянии, или требует ремонта стоимости w_i. Ездить можно только по дорогам в рабочем состоянии. За минимальную стоимость отремонтировать некоторые дороги так, чтобы из любого города страны можно было добраться в любой другой.

2. **(2+1) Кратчайший путь**

   Дан неориентированный взвешенный граф. Сделайте за $\mathcal{O}(E \log V)$ предподсчёт, и научитесь затем для любой пары вершин возвращать путь между a_i и b_i, в котором минимальный вес ребра максимален.

   a) **(2)** За $\mathcal{O}(V)$.
   b) **(+1)** За $\mathcal{O}$(**размера ответа**).

   Заметьте, что найти путь за нужное время не так уж и просто!

3.1. Дополнительная часть

1. **(3) Заправки**

   Страна – города и односторонние дороги. У нас есть машина с баком бензина размера C, в некоторых городах есть заправки с бесконечным бензином. Для каждой дороги известно, сколько бензина нужно потратить, чтобы по ним проехать. Мы стартуем в городе s с полным баком. До куда мы сможем доехать?

   a) **(1)** $\mathcal{O}(VE)$
   b) **(+2)** $\mathcal{O}(E \log V)$