SPb HSE, ПАДИИ, 1 курс, весна 2023/24
Практика по алгоритмам #5

bfs/dijkstra

15 февраля

Собрано 15 февраля 2024 г. в 10:54

Содержание

bfs/dijkstra

1. **Чётные пути**

   Дан орграф, $w_e \geqslant 0$. Найдите кратчайший путь из s в t, состоящий из чётного числа рёбер.

2. **Кратчайший путь по выделенным вершинам**

   Дан взвешенный орграф с положительными весами.
   Найти кратчайший путь из s в t, проходящий по всем $k \leqslant 10$ выделенным вершинам.

3. **Дейкстра и `C++`**

   `set` или `priority_queue`?
   `pair<int,int>` или свой тип?
   Как передать свой тип?

4. **Число кратчайших путей**

   Дан взвешенный орграф с положительными весами и вершина s.
   Для каждой вершины v найти число кратчайших путей из v в s.
   Для каждого ребра проверить, $\exists$ ли кратчайший путь через e

5. **1-k-bfs**

   Дан граф, в котором все веса целые и имеют значения от 1 до k. Решите SSSP:

   a) $\mathcal{O}(k(V+E))$
   b) $\mathcal{O}(kV+E)$
   c) То же с вещественными весами.

6. **Видим графы**

   a) Ряд из 16 клеток. В каждой клетке изначально один монстр, у каждого сила a_i. За ход можно из двустволки зачистить две соседние клетки. После этого «ходят» монстры, мы получаем ущерб $\sum a_i$ выживших. Зачистить монстров, минимизировать ущерб.
   b) На плоскости 100 прямоугольных непересекающихся препятствий. Найти кратчайший путь из x_1, y_1 в x_2, y_2.
   c) Дороги бывают платные. Дойти за минимальное время из s в t, использовав не более 100 платных дорог (бюджет).

7. **Кратчайший цикл**

   Найти кратчайший цикл в (a) орграфе, (b) неорграфе.

8. **Самый длинный цикл**

   Найти в неорграфе самый длинный цикл. За сколько можно решать такую задачу?

Разбор задач практики

1. **Чётные пути**

   Состояние $= (v, b)$, где $v \in V$, $b \in \{0, 1\}$. b отвечает за чётность длины пути в v.
   Ищем путь из $(s, 0)$ в $(t, 0)$.

2. **Кратчайший путь по выделенным вершинам**

   Запустим Дейкстру из k фиксированных вершин и начальной.
   Теперь сделаем орграф только из k+2 вершин – фиксированные, начальная, конечная.
   Вес ребер в нем равен кратчайшему пути в исходном графе.
   Получили задачу коммивояжёра на k+2 вершинах. Итого $\mathcal{O}(k \cdot \texttt{Dijkstra} + 2^k k^2)$.

   Альтернативное решение. Сразу пустить Дейкстру на графе из $2^k n$ вершин,
   где вершина = пара $\langle$маска посещённых выделенных, текущая$\rangle$.

3. **Дейкстра и** `C++`

   `set<pair<int,int>> s;` – красно-чёрное дерево пар.
   `priority_queue<pair<int,int>> q;` – куча пар.
   Почему пар? Добавляем вершину v:
   `s.insert({d[v], v})`, `q.push({-d[v], v})`: сортируются по возрастанию `d[v]`.
   Почему `-d[v]`? Иначе было бы по убыванию.

   Свой тип: `struct T { int id };`
   Чтобы можно было создавать `set<T>`, нужно определить `operator<`:

```
bool operator<(T a, T b) {
    return make_pair(d[a.id],a.id) < make_pair(d[b.id],b.id);
}
```

   Зачем `make_pair`? Чтобы все вершины были различны, `set`-у сложно будет удалять, если есть одинаковые. Теперь можем писать `set<T> s; s.insert({start});`

   Для `priority_queue` достаточно оператора `return d[a.id] > d[b.id];`

4. **Число кратчайших путей**

   Решение #1.
   Запускаем Дейкстру, оставляем только ориентированные ребра (a, b, w): $d[a] + w = d[b]$.
   Получили так называемый «граф кратчайших путей».
   $\forall e$: $w[e] > 0 \Rightarrow$ граф кратчайших путей ацикличен. Считаем на нём динамикой число путей из s во все v.
   Если есть нулевые ребра, будут нулевые циклы, по которым можно бесконечно крутиться.

   Решение #2: по ходу Дейкстры считаем еще и $\texttt{count}[v]$ – число путей длины $d[v]$.

5. **Модификации bfs**

   a) *1-k-bfs*
      Можно разбить ребро на k кусков длины 1.
      Можно создать kV очередей, в i-й хранить вершины v: $d[v] = i$.
      Подробно: см. конспект пми.

b) *1-2-bfs, веса вещественные.*
Три очереди: вершины на расстояниях $[d, d+1)$, $[d+1, d+2)$, $[d+2, d+3)$.
Когда вынимаем из первой очереди вершину на расстоянии d' и смотрим из нее на ребро веса w, будет $d' + w < d + 3$. По $d' + w$ поймем, в какую очередь ее положить.

Корректность, как у 1-k-bfs. Инвариант: на текущий момент верно посчитаны расстояния для всех вершин на расстоянии $< d + 1$.
Когда опустела $[d, d+1)$ очередь, инварант верен уже для следующей очереди: если до вершины расстояние $[d+1, d+2)$, то до предыдущей на кратчайшем пути оно $[d-1, d+1)$, такие уже обработаны.

6. **Видим графы**

a) Монстры: граф на 2^16 вершинах – живые монстры. Дейкстра.

b) Прямоугольники: граф на $100 \times 4 + 2$ вершинах – в каком углу прямоугольника стоим. Рёбра – прямые отрезки. Ребро проводим, если оно ничего не пересекает.

c) Платные дороги: вершина $\langle v, k \rangle$ – стоим в v, прошли уже k платных дорог.

7. **Кратчайший цикл**

Орграф: `bfs` из каждой вершины, смотрим первый момент, когда вернулись в стартовую вершину. $\mathcal{O}(VE)$.

Неорграф. Из каждой вершины запускаем `bfs`. Каждое не древесное ребро образует цикл. $\mathcal{O}(VE)$. Когда мы запустим `bfs` из вершины цикла, мы найдём кратчайший цикл. Если это не простой цикл, то существует цикл короче, противоречие.

8. **Самый длинный цикл**

Если есть гамильтонов цикл, он и есть ответ. Так что мало шансов решить такую задачу за полином. Умеем за $\mathcal{O}(2^n n)$.

Домашнее задание

3.1. Обязательная часть

1. **(3) Число кратчайших путей**

   Дан невзвешенный неорграф и два множества вершин – A и B, $A \cap B = \varnothing$. Найти

   a) (длину кратчайшего пути из A в B) за $\mathcal{O}(n+m)$.
   b) (количество кратчайших путей из A в B) mod 10^9 за $\mathcal{O}(n+m)$.

3.2. Дополнительная часть

1. **(3) Сумма двух максимальных ребер**

   Есть взвешенный неограф с положительными весами рёбер. Найти путь из s в t такой, что сумма двух максимальных ребер на пути минимальна. $\mathcal{O}((n+m) \cdot \text{poly}(\log n))$.