Алгоритмы и структуры данных. Часть 2. Лекция 1

Конспекты

10 февраля

Содержание

1. Графы с отрицательными ребрами

1.1. Форд-Белман с очередью

Пока можем улучшать ребро, улучшаем

```
while Run
  Run ← 0
  for e : a → b
    if d_a + w_ab < d_b
      d_b ← a_a + w_ab
      Run ← 1
```

1. очередь с верщиными для которых поменялось расстояние

   Примечание: Если внешний цекл сделал $> n - 1$ итераций, то есть цикл отрацательного веса.

Теорема 1.1.1: Эта штука работате за $O(n \cdot m)$.

Доказательство: Пусть depth — глубина дерева минимальных путей. Тогда время работы $\text{Time} = E \cdot \text{depth}$. Где depth рандомный в среднем $\log n$. ■

1.2. Через динамику

dp[k, v] — минимальный вес пути. Пересчитывем переход $k \mapsto k + 1$ за $\mathcal{O}(E)$.

Примечание:

$$\forall e : v \to u \quad \text{dp}[k+1, v] = \min(\text{dp}[k+1, v], \text{dp}[k, v] + w_{vu})$$

1.3. Отрицательные циклы

Примечание: Если делаем Форда-Белмана на очереди, то если сделали n итераций то существует отрицательный цикл. Однако нельзя сказать что если улучшили на последней итерации расстояние до v, то не значит что она в цикле, она достижима из него. Поэтому чтобы найти сам цикл надо пройти по дереву обхода обратно. Если найдем какой-то цикл то это и есть он.

Доказательство: Предположим что не закциклились. Тогда есть путь от s до v. Рассмторим какое-то $p[a]$ где $e : a \to \cdot$.

$$d[p[a]] + w_e \le d[a] \Longrightarrow \underbrace{d[s]}_{0} + \underbrace{\omega(\text{path})}_{\le n-1} \le \underbrace{d[v]}_{n\text{-тая фаза}}$$

Покажем что этот цикл отрицательный. **Доделать**

$$\sum d + \omega(\text{cycle}) \le \sum d$$

∎

1.4. Флойд

$d[i, j] = \begin{cases} w_{ij} \\ +\infty \end{cases}$

for k
- **for** i
  - **for** j
    - $d[i, j] \overset{\text{Relax}}{\leftarrow} d[i, k] + d[k, j]$

Доказательство:
- $k = 0$. Очев
- $k \mapsto k + 1$. Уже знаем минимум для всех k. Очев? **Но это не точно**

∎

Примечание:
- Отрицательные циклы если есть $d[u, v] < 0$
- Восстановление пути. Храним следующую вершину в пути $p[i, j]$. Изначально $p[i, j] = j$. Если обновляем через k, то $p[i, j] = p[i, k]$. Потом идем по $i = p[i, j]$.

1.5. Потенциалы

Можем поменять веса ребер так для какого-то $e : a \to b$. $w_e \mapsto w_e + p[a] - p[b]$. Посмотрим как изменятся веса путей $w(\text{path}) \mapsto w(\text{path}) + p[s] - p[t]$. Где p — **потенциалы** — любые вещественные числа. Можно заметить что минимальный путь останется минимальным.

Как подобрать p так чтобы веса стали неотрицательными. Найдем расстояние из фиктивной вершины до всех остальных вершин Форд-Белманом. Возьмем расстояния d как потенциалы.

Доказательство: Покажем что $w_e + d[a] + d[b] \ge 0$. Если оказалось что $w_e + d[a] < d[b]$ то это значит что $d[b]$ посчитано неверно. Противоречие. ∎

1.5.1. Джонсон

Чтобы найти расстояния от каждой вершины до каждой то Флойдом V^3. Но можем Ф.Б найти потенциалы, избавиться от отрицательных ребер и запустить Дейкстру n раз: $\text{F.B.} + V \cdot \text{Dijkstra} - VE + V^2 \log V \le V^3$

1.6. Транизитивное замыкание

Можно решать Флойдом:

for k
- **for** i
  - **for** j
    - **if** $c_{ik} \land c_{kj}$
      - $c_{ij} \leftarrow T$

Пооптимизируем. Заметим что делаем `c[j] |= c[k]`. Можно сделать на битсетах.