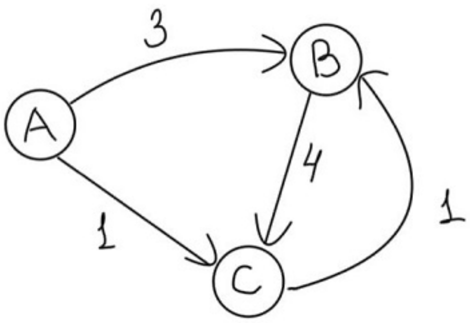
Комбинаторика и теория графов. Лекция 9

Шаруева Полина Петровна студентка группы 0371

Алгоритм Форда-Беллмана



A: 3B, 1C
B: 4C
C: 1B

пути из A:

1 шаг - релаксируем:
длина пути из A в B: 3
$0+3 < \infty \Rightarrow$ записываем в таблицу

2 шаг - релаксируем:
длина пути из A в C: 1
$0+1 < \infty \Rightarrow$ записываем в таблицу

3 шаг:
длина пути из B в C: 4
$4+3 < 1$ - не верно $\Rightarrow$ не записываем в таблицу

4 шаг - релаксируем:
длина пути из C в B: 3
$1+1 < 3 \Rightarrow$ записываем в таблицу

A	B	C
0	∞	∞
0	3	∞
0	3	1
0	2	1

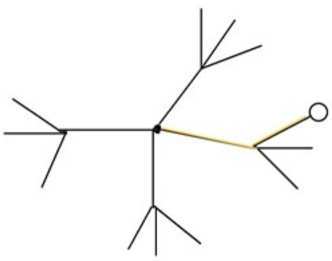
n = 3 $\Rightarrow$ n - 1 = 2 раза цикл релаксации
AB, AC, BC, CB - нет улучшений

Ответ: A - 0, B - 2, C - 1

Алгоритм Форда-Беллмана

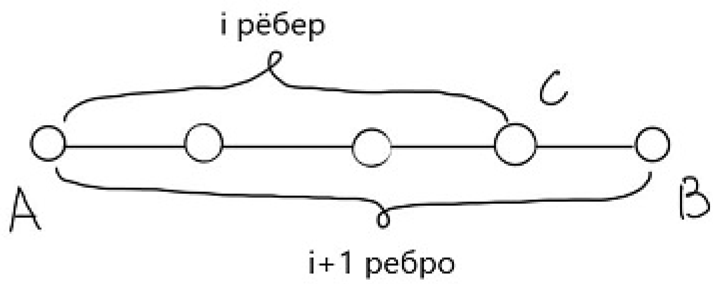
Теорема: В конце массива d содержит расстояния от A

Доказательство: После i-го релаксации всех рёбер, d хранит $d(v) \leqslant$ min длин путей, в которых $\leqslant$ i рёбер



Действительно. *База:* $i = 0$
min (пути из 0 рёбер) - только A-A
$d(A) = 0$
$d(u) = \infty$

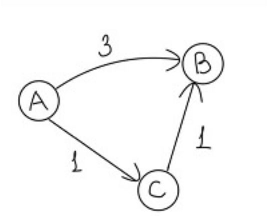
Переход: $\sqsupset$ есть оптимальный путь из i+1 ребра



По предположению:
$d(C) = d(B) + (A, C)$

- длина пути A - C - B = dist(C) + вес(CB)
*d(C) = dist(C)

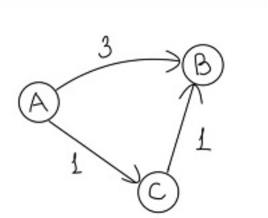
Проверка:
$d(C) +$ вес$(CB) \leqslant d(B)$ - верно, т.к. путь оптимальный
$\Rightarrow d(B) = d(C) +$ вес(CB)



(путь после второй релаксации будет равен 2)

Почему n - 1 этап?

Оптимальный путь не содержит цикл



$\Rightarrow$  $\leqslant$ n-1 ребро. Ч.т.д.

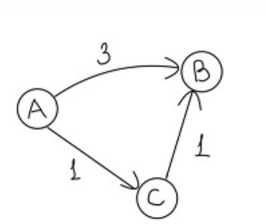
Замечание:

Мы вычисляем только расстояния, но путь неизвестен. Как восстанавливать пути?

Будем сохранять информацию об успешных релаксациях
Prev - массив вершин

Если релаксация 

успешная, то Prex[v] = u (оптимальный путь в v лежит через u)



	A	B	C
d	0	∞	∞
AB:	0	3/A	∞
AC:	0	3/A	1/A
CB:	0	2/C	1/A

Восстанавливаем путь в B
A $\rightarrow$ C $\rightarrow$ B
* A - prev(C) ; C - prev(B)

В общем случае путь A $\rightarrow$ v это:
A $\rightarrow$ ... $\rightarrow$ prev(prev) $\rightarrow$ prev(v) $\rightarrow$ v

Алгоритм Дейкстры

В отличие от ФБ требует, чтобы все веса w(e) $\geqslant$ 0

Алгоритм:

Дан граф G = (V, E), A $\in$ V

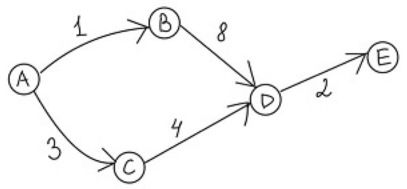
Найти расстояние до всех вершин d(u) = dist(A, u)

d(A) = 0 d(u $\neq$ A) = ∞ - обработанные вершины

Повторяем n раз: (n = V) - вершины

Выбрать u $\in$ V $\setminus$ P, где d(u) $\rightarrow$ min (из необработанных min(d))
for e $\in$ ребра из U, e(u, v)
релаксируем ребро e
P = P_v {U}

Пример:

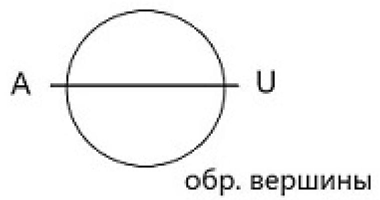


	A	B	C	D	E
d	0	∞	∞	∞	∞
		1	3	∞	∞
			3	9	∞
				7	∞
					9

Эффективность: lVl * lEl * loglVl

Корректность:

Идея: на каждом шаге d(u) = min путей



База:

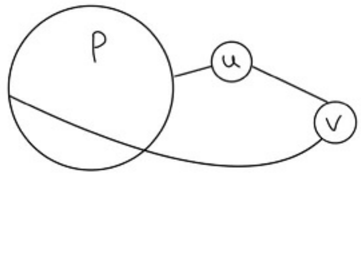
Шаг = 0 d(A) = 0 d(u) = ∞

Переход:

Выбираем u = min вершин из V $\setminus$ {P}

$\sqsupset$ есть оптимальный путь в u: A - ... - $\overline{u}$ - ... - u

По предположению,
dist($\overline{u}$) = d($\overline{u}$)
$\uparrow$
dist(u)
$\uparrow$
d(u) $\Rightarrow$ d(u) $>$ d($\overline{u}$) $\Rightarrow$?? - противоречие
d(u) был min



$\sqsupset$ оптимальный путь в V идет через u $\mathrm{dist}(A, u) = w(u, v) = \mathrm{dist}(A, v)$

$\Rightarrow$ релаксация $u \to v$ успешная, и $d(v)$ получит расстояние. Ч.т.д.

Для восстановления пути.

Нужен аналогичный Prev.

- успешная релаксация
Prev[v] = u

	A	B	C	D	E
d	0	∞	∞	∞	∞
		1/A	3/A	∞	∞
			3/A	9/B	∞
				7/C	∞
					9/D

путь A - E

$A \to C \to D \to E$

*A - prev(C), C - prev(D), D - prev(E)