Кратчайшие пути между всеми парами вершин

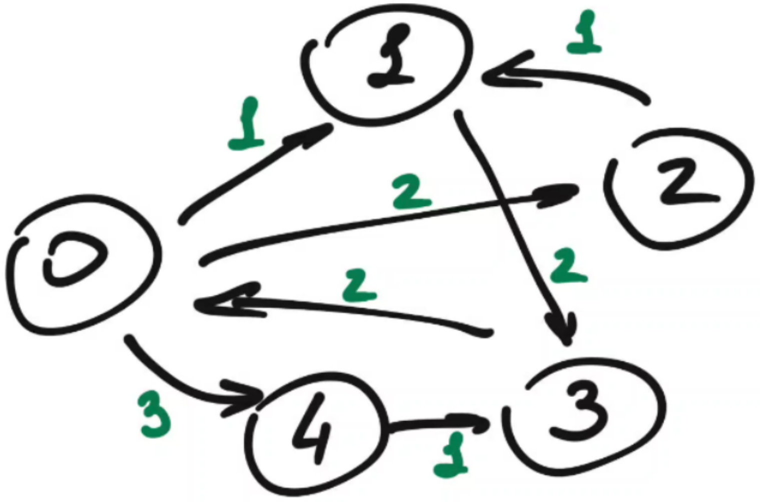
All-Pairs Shortest Paths

Кратчайшие пути между всеми парами вершин

Дан граф $G = (V, E)$, на котором задана весовая функция $w : E \to \mathbb{R}$.

Предположим отсутствие отрицательных циклов. Для любой пары $v, u \in V$ требуется найти $\rho(v, u) = \min\limits_{P} \sum\limits_{e \in P(v,u)} w(e)$, где $P(v, u)$ - путь из v в u.

То есть в качестве ответа ожидается матрица d: $d[v][u] = \rho(v, u)$.



from \ to	0	1	2	3	4
0	0	1	2	3	3
1	4	0	6	2	7
2	5	1	0	3	8
3	2	3	4	0	5
4	3	4	5	1	0

Наивный подход

Уже знаем, как искать КП из одной вершины.

Запустим эти алгоритмы для каждой вершины (V раз).

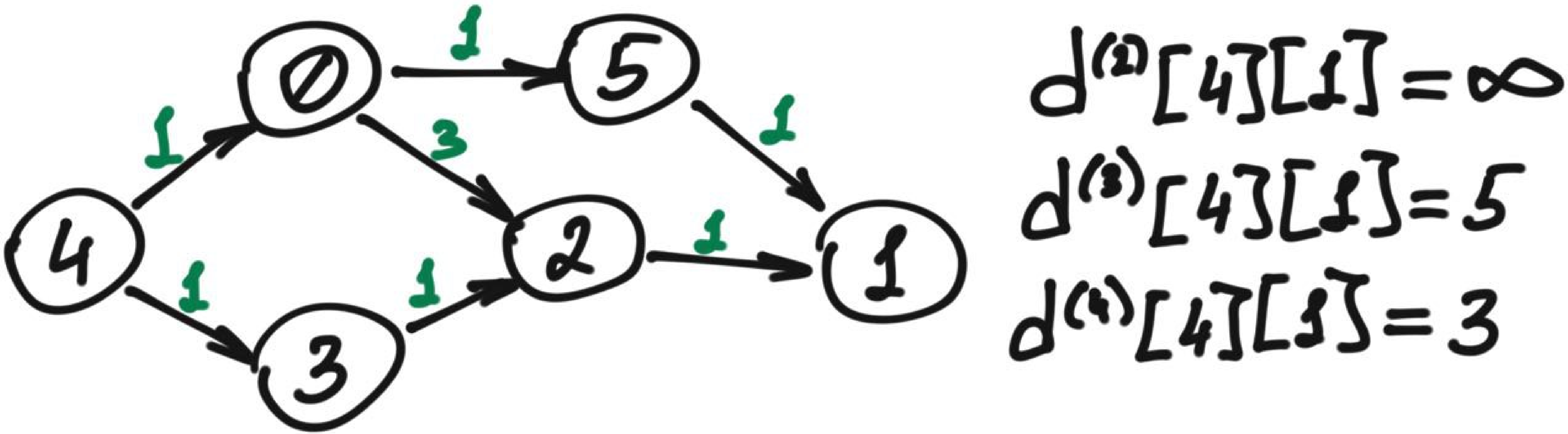
	$w \in \mathbb{R}$	$w \geq 0$
$E \sim V$	ФБ: $O(V^3)$ Дейкстра: — • •	$O(V^3)$ $O(V^2 \log V)$
$E \sim V^2$	ФБ: $O(V^4)$ Дейкстра: — • •	$O(V^4)$ $O(V^3)$

Алгоритм Флойда-Уоршелла

Алгоритм Флойда-Уоршелла

Воспользуемся методом динамического программирования.

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$ (в 0-индексации).



Алгоритм Флойда-Уоршелла: динамика

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$.

- Чему равен $d^{(0)}[x][y]$?

Алгоритм Флойда-Уоршелла: динамика

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$.

- $d^{(0)}[x][y] = w(x, y)$

Чему равен $d^{(k+1)}[x][y]$, если известно, что этот КП НЕ идет через вершину k?

Алгоритм Флойда-Уоршелла: динамика

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$.

- $d^{(0)}[x][y] = w(x, y)$
- Если КП из x в y по вершинам $< k+1$ не проходит через k, то $d^{(k+1)}[x][y] = d^{(k)}[x][y]$

Чему равен $d^{(k+1)}[x][y]$, если известно, что этот КП ИДЕТ через вершину k?

Алгоритм Флойда-Уоршелла: динамика

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$.

- $d^{(0)}[x][y] = w(x, y)$
- Если КП из x в y по вершинам $< k+1$ не проходит через k, то $d^{(k+1)}[x][y] = d^{(k)}[x][y]$
- Если КП из x в y по вершинам $< k+1$ проходит через k, то $d^{(k+1)}[x][y] = d^{(k)}[x][k] + d^{(k)}[k][y]$

Чему равен $d^{(k+1)}[x][y]$ в общем случае?

Алгоритм Флойда-Уоршелла: динамика

Пусть $d^{(k)}[x][y]$ - кратчайшее расстояние от x до y вдоль пути, в котором в качестве промежуточных используются лишь вершины с номерами $< k$.

- $d^{(0)}[x][y] = w(x, y)$
- Если КП из x в y по вершинам $< k+1$ не проходит через k, то $d^{(k+1)}[x][y] = d^{(k)}[x][y]$
- Если КП из x в y по вершинам $< k+1$ проходит через k, то $d^{(k+1)}[x][y] = d^{(k)}[x][k] + d^{(k)}[k][y]$
- $d^{(k+1)}[x][y] = \min(d^{(k)}[x][y], d^{(k)}[x][k] + d^{(k)}[k][y])$

$$\min\left(x \xrightarrow{<k} y,\ x \xrightarrow{<k} k \xrightarrow{<k} y\right)$$

Алгоритм Флойда-Уоршелла: реализация

```
def FloydWarshall(W):  # W - матрица весов ребер
  d[0] = W
  for k from 0 to V - 1:
    for x from 0 to V - 1:
      for y from 0 to V - 1:
        d[k + 1][x][y] = min(d[k][x][y], d[k][x][k] + d[k][k][y])
  return d[V]
```

Сложность: $T(V, E) = \Theta(V^3)$, $M(V, E) = \Theta(V^3)$

Замечание: нельзя изменять положение цикла по k!

Алгоритм Флойда-Уоршелла: версия 2.0

Заметим, что в каждый момент времени достаточно хранить две матрицы.

```
def FloydWarshall(W):  # W - матрица весов ребер
  d_old = d_new = W
  for k from 0 to V - 1:
    swap(d_old, d_new)
    for x from 0 to V - 1:
      for y from 0 to V - 1:
        d_new[x][y] = min(d_old[x][y], d_old[x][k] + d_old[k][y])
  return d_new
```

Сложность: $T(V, E) = \Theta(V^3)$, $M(V, E) = \Theta(V^2)$

Замечание: нельзя изменять положение цикла по k!

Алгоритм Флойда-Уоршелла: версия 3.0

На самом деле, можно реализовывать алгоритм in-place.

```
def FloydWarshall(d):  # d - матрица весов ребер
  for k from 0 to V - 1:
    for x from 0 to V - 1:
      for y from 0 to V - 1:
        d[x][y] = min(d[x][y], d[x][k] + d[k][y])
  return d
```

Сложность: $T(V, E) = \Theta(V^3)$, $M(V, E) = \Theta(1)$

Замечание: нельзя изменять положение цикла по k!

Алгоритм Флойда-Уоршелла: версия 3.0

На самом деле, можно реализовывать алгоритм in-place.

```
def FloydWarshall(d):  # d - матрица весов ребер
  for k from 0 to V - 1:
    for x from 0 to V - 1:
      for y from 0 to V - 1:
        d[x][y] = min(d[x][y], d[x][k] + d[k][y])
  return d
```

Почему так можно?

Потенциально, может быть проблема "заглядывания" в будущее, например:
$d^{(k+1)}[x][y] = d^{(k+1)}[x][k] + d^{(k)}[k][y]$.

Но очевидно, что при отсутствии циклов отрицательного веса,
$d^{(k+1)}[x][k] = d^{(k)}[x][k]$ и $d^{(k+1)}[k][y] = d^{(k)}[k][y]$.

Поэтому все ок.

Отрицательные циклы

А что там с отрицательными циклами? Возможно ли определить их наличие?

Отрицательные циклы

Отрицательный цикл = найден пут отрицательного веса из вершины в нее же.

Таким образом, если на диагонали появились отрицательные значения, значит есть цикл отрицательного веса.

```
def FloydWarshall(d): # d - матрица весов ребер
  for k from 0 to V - 1:
    for x from 0 to V - 1:
      for y from 0 to V - 1:
        d[x][y] = min(d[x][y], d[x][k] + d[k][y])

  for i from 0 to V - 1:
    if d[i][i] < 0:
      NegativeCycle!
```

Промежуточный итог

		$w \in \mathbb{R}$	$w \geq 0$
$E \sim V$	ФБ:	$O(V^3)$	$O(V^3)$
	Дейкстра:	—	$O(V^2 \log V)$
	Флойд:	$O(V^3)$	$O(V^3)$
	•		
$E \sim V^2$	ФБ:	$O(V^4)$	$O(V^4)$
	Дейкстра:	—	$O(V^3)$
	Флойд:	$O(V^3)$	$O(V^3)$
	•		

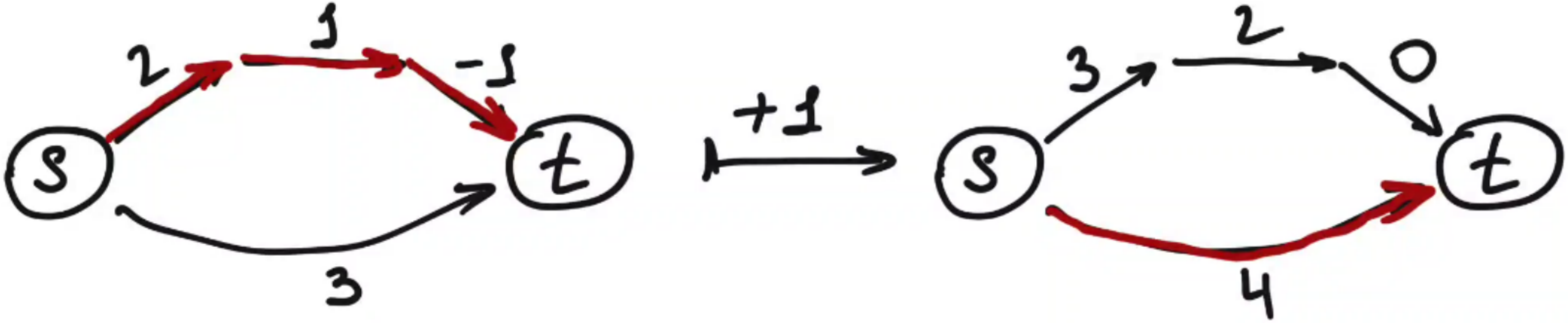
Метод потенциалов. Алгоритм Джонсона.

Задача

Хотим применить алгоритм Дейкстры, но в графе есть отрицательные ребра.

Попробуем изменить веса ребер так, что отрицательных ребер не останется, но при этом кратчайшие пути не изменятся.

Наивный подход со сдвигом весов не работает:



Метод потенциалов

Пусть $\varphi : V \to \mathbb{R}$ некоторая функция на вершинах (потенциал).

Определим новую весовую функцию на ребрах
$w_\varphi(x, y) = w(x, y) + \varphi(x) - \varphi(y)$

- Как изменится длина **произвольного** пути из v в u?

Метод потенциалов

Пусть $\varphi : V \to \mathbb{R}$ некоторая функция на вершинах (потенциал).

Определим новую весовую функцию на ребрах
$w_\varphi(x, y) = w(x, y) + \varphi(x) - \varphi(y)$

- $w_\varphi(P(v, u)) = w(P(v, u)) + \varphi(v) - \varphi(u)$
- Изменится ли кратчайший путь из v в u?

Метод потенциалов

Пусть $\varphi : V \to \mathbb{R}$ некоторая функция на вершинах (потенциал).

Определим новую весовую функцию на ребрах
$w_\varphi(x, y) = w(x, y) + \varphi(x) - \varphi(y)$

- $w_\varphi(P(v, u)) = w(P(v, u)) + \varphi(v) - \varphi(u)$
- Кратчайший путь из v в u **не изменится**, только его вес (на $\varphi(v) - \varphi(u)$).
- Как изменится вес любого цикла?

Метод потенциалов

Пусть $\varphi : V \to \mathbb{R}$ некоторая функция на вершинах (потенциал).

Определим новую весовую функцию на ребрах
$w_\varphi(x, y) = w(x, y) + \varphi(x) - \varphi(y)$

- $w_\varphi(P(v, u)) = w(P(v, u)) + \varphi(v) - \varphi(u)$
- Кратчайший путь из v в u **не изменится**, только его вес (на $\varphi(v) - \varphi(u)$).
- Веса циклов **не меняются**.

Метод потенциалов

Пусть $\varphi : V \to \mathbb{R}$ некоторая функция на вершинах (потенциал).

Определим новую весовую функцию на ребрах
$w_\varphi(x, y) = w(x, y) + \varphi(x) - \varphi(y)$

- $w_\varphi(P(v, u)) = w(P(v, u)) + \varphi(v) - \varphi(u)$
- Кратчайший путь из v в u **не изменится**, только его вес (на $\varphi(v) - \varphi(u)$).
- Веса циклов **не меняются**.

Вывод: если в исходном графе нет отрицательных циклов, то и в измененном их нет. И все кратчайшие пути совпадают с кратчайшими путями в исходном графе.

Осталось придумать потенциал, который избавит нас от отрицательных весов.

Метод потенциалов

Добавим в граф G фиктивную вершину s^*, из которой исходят ребра веса 0 во все остальные вершины.

Запустим алгоритм Форда-Беллмана и найдем кратчайшие расстояния от s^* до остальных вершин.

Теорема (D.B.Johnson, 1977).

Пусть в графе G отсутствуют циклы отрицательного веса, и $\varphi(\cdot) = \rho(s^*, \cdot)$. Тогда $\forall (v, u) \in E : w_\varphi(v, u) \geq 0$

Метод потенциалов

Добавим в граф G фиктивную вершину s^*, из которой исходят ребра веса 0 во все остальные вершины.

Запустим алгоритм Форда-Беллмана и найдем кратчайшие расстояния от s^* до остальных вершин.

Теорема (D.B.Johnson, 1977).

Пусть в графе G отсутствуют циклы отрицательного веса, и $\varphi(\cdot) = \rho(s^*, \cdot)$. Тогда $\forall (v, u) \in E : w_\varphi(v, u) \geq 0$

Доказательство.

$w_\varphi(v, u) := w(v, u) + \varphi(v) - \varphi(u) = w(v, u) + \rho(s^*, v) - \rho(s^*, u) \geq 0$

Последнее неравенство верно, так как $\rho(s^*, v) + w(v, u) \geq \rho(s^*, u)$ (путь из s^* в u, проходящий по ребру (v, u), не короче кратчайшего пути). ■

Алгоритм Джонсона

1. Добавляем фиктивную вершину s^* с нулевыми ребрами во все вершины.
2. Ищем кратчайшие пути $\forall x : \rho(s^*, x)$ с помощью алгоритма Форда-Беллмана.
3. $\forall x : \varphi(x) := \rho(s^*, x), \forall (v, u) \in E : w_\varphi(v, u) := w(v, u) + \varphi(v) - \varphi(u)$.
4. С помощью алгоритма Дейкстры ищем кратчайшие пути из каждой вершины.

Сложность алгоритма $O(VE + VE \log V)$ или $O(VE + V(E + V^2))$ в зависимости от реализации алгоритма Дейкстры.

То есть для разреженных графов - $O(V^2 \log V)$, для плотных графов - $O(V^3)$.

Итог

		$w \in \mathbb{R}$	$w \geq 0$
$E \sim V$	ФБ:	$O(V^3)$	$O(V^3)$
	Дейкстра:	—	$O(V^2 \log V)$
	Флойд:	$O(V^3)$	$O(V^3)$
	Джонсон:	$O(V^2 \log V)$	—
$E \sim V^2$	ФБ:	$O(V^4)$	$O(V^4)$
	Дейкстра:	—	$O(V^3)$
	Флойд:	$O(V^3)$	$O(V^3)$
	Джонсон:	$O(V^3)$	—