Теория графов

silvia.lesnaia

7 ноября 2025 г.

19.19.25

1 Графы знаний

Представление знаний - направление в исследованиях ИИ, посвященное представлению информации о мире в форме, которую было бы возможно использовать в компьютерных...

Социальные графы

Молекулярные графы

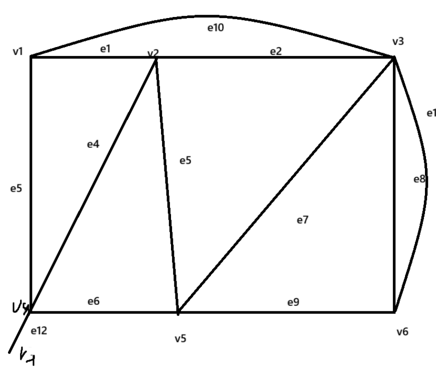
Графы знаний

Задачи на уровне всего графа

Задачи на уровне вершин

Задачи на уровне ребер

уту много всего надо запоннить



Маршурт

$V2 \to V6$

110311

...

Расстояние d(x,y), между двумясвязаными вершинами x,y назвается длина наименьшого пути между нимми

2 Алгоритм обходов

24.10.25

Мария Салдина

тут картинки должны быть. Вставить по теме

...

Анастасия Гожева

B.B quee цвета - HHHHHHHHHH 1 11HHHHHHH

3 4 12HHHH

4 5 12 11HH

5 6 121122

6 121122

Проверка графа на двудольность

тут картинка

Проверка на ацикличность

Виктория Лазарева - заменяет лектора

3 Алгоритм Форда-Беллмана

Внешний граф -

В этой теме рассматривается внещний граф.

Картинки тут тоже

31.10.25

4 Алгоритм Дейкстры: поиск кратчайших путей от одной вершины

Задача: Дан взвешенный граф $G = (V, E)$. Веса рёбер $w(u, v) \geq 0$ (неотрицательные). Найти кратчайшие пути от заданной исходной вершины s до всех остальных вершин графа.

Неопходимые определения

- **Вес пути:** Сумма весов всех рёбер, входящих в этот путь.
- **Кратчайший путь ($p(s, v)$):** путь минимально возможным весом из s в v.
- **Массив dist[], где** $dist[v]$ — длина кратчайшего пути из s в v.
- **Массив prev[],** позволяющий восстанавливать сам путь.

- **Операция «Релаксации»:** Основная операция.
  - Имеем: текущая оценка расстояния до v — $dist[v]$.
  - Рассматриваем ребро (u, v).
  - Если $dist[u] + w(u, v) < dist[v]$, то мы нашли более короткий путь через u.
  - Обновляем: $dist[v] = dist[u] + w(u, v)$ и $prev[v] = u$.

Описания алгоритма

5 Алгоритм Флойда-Уоршелла

Проблема: Дан взвешенный граф. Необходимо найти кратчайшие пути между вершинами.

Ключевые вопросы: Из одной вершины:

1 Найти кратчайшие пути от одной выделенной вершины до всех остальных.

2 Между всеми парами: Найти кратчайшие пути от каждой вершины до каждой.

Определения

Граф: G = (V, E), где

V — множество вершин (|V| = n), E — множество рёбер (|E| = m).

Взвешенный граф: Каждому ребру (u, v) сопоставлен вес (или длина) w(u, v).

Кратчайший путь между вершинами i и j: Путь, сумма весов рёбер которого минимальна среди всех возможных путей из i в j.

Расстояние: Длина кратчайшего пути между двумя вершинами. Обозначение d(i, j)

Важно: Алгоритм Флойда-Уоршелла работает с ориентированными и неориентированными графами. Допускает отрицательные веса, но не допускает циклов отрицательного веса. Граф будет взвешанный

Алгоритм

Вход:

Взвешенный граф с n вершинами.

Матрица смежности W размером n x n, где:

W[i][j] = w(i, j) = вес ребра (i, j), если оно существует.

W[i][j] = ∞ (большое число), если ребра нет.

W[i][i] = 0 (расстояние от вершины до самой себя).

Выход:

Матрица D размером n x n, где D[i][j] = d(i, j) — длина кратчайшего пути из i в j.

(Необязательно) Матрица предшествования P для восстановления самих путей.

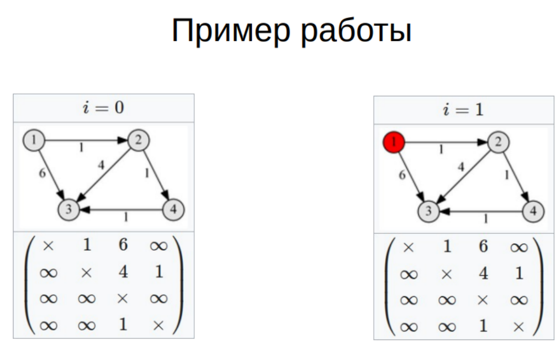
Псевдокод

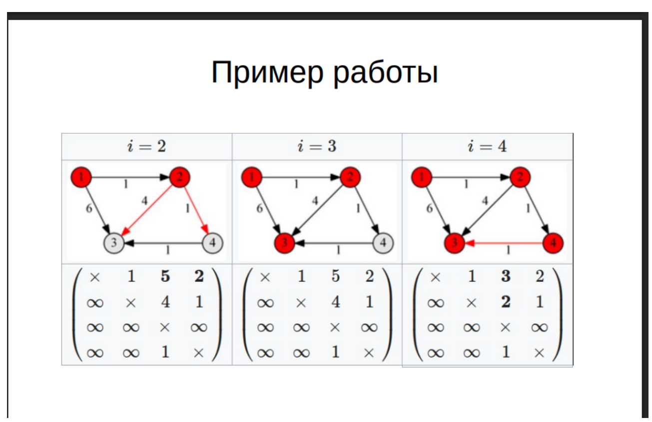
1.for each vertex k in V do

2. for each vertex i in V do
3. for each vertex j in V do
4. if dist[i][k] + dist[k][j] < dist[i][j] then
5. dist[i][j] := dist[i][k] + dist[k][j]

Общая сложность — O(|V|3), что обусловлено тремя вложенными циклами.

Потребление памяти — O(|V|2) для хранения матрицы расстояний.





Характеристика

Плюсы: 1. Простота реализации (всего 3 цикла).

2. Работает с отрицательными весами (но без отрицательных циклов).

3. Находит расстояния между всеми парами вершин.

Минусы:

1. Кубическое время делает его непрактичным для очень больших графов.

2. Для задачи "из одной вершины"алгоритм Дейкстры (работающий за O(|E| + |V| log |V|)) значительно эффективнее.

Примеры задач

Поиск кратчайших путей в фундаментальных задачах маршрутизации

Анализ сетей и маршрутов

Определение возможности негативных циклов

Области применения

Транспортное планирование: Расчет кратчайших маршрутов между всеми парами городов.

Анализ сетей: Построение матрицы достижимости в графе (если нас интересует только факт существования пути, веса можно положить равными 1).

Маршрутизация в компьютерных сетях: Протоколы, подобные OSPF, используют идеи, близкие к алгоритму ФлойдаУоршелла.

Поиск транзитивного замыкания графа: Модификация алгоритма позволяет определить, достижима ли вершина j из вершины i (алгоритм Уоршелла).

6 Связующие дерево

Остовные деревья Обычно нам нужно добраться из любой вершины в любую другую и использовать как можно меньше ребер. При этом часто возникает проблема поиска связного подграфа графа, в котором используется как можно меньше ребер. Чтобы избежать этой проблемы, нужно использовать связные деревья.

Рассмотрим следующую задачу:

Авиакомпания содержит m рейсов между n городами, i -ый из них обходится в wi рублей, причём из любого города можно добраться до любого другого. В стране наступил кризис, и нужно отказаться от как можно большего числа из них таким образом, что содержание оставшихся рейсов будет наиболее дешёвым.

07.11.25

7 Алгоритм Джонсона

...

Алгоритмическая сложность

тут все вствaить из презентации

Общая сложность Джонсона

Эффективн для разреженных грфоф.

Выбор алгоритма

8 Сети. Потоки в сетях

пропускная способность

Форд-Фалкерсон

тут тоже

Алгоритм масштабирования потока тут текст тут текст

Псевдокод

func maxScale(G,S,t)

выст 0 $\bigvee(v, V)$

$sscale = 2^{xx} floor(bq_2(U))$

$novaside > 1$

$deltamin(c_f)(u, v)(u, v)u, v$

$f(u, v) := f(u, v) + delta$

$f(v, u) := f(v, u) - delta$

$scale = scale//2$