Алгоритм Уоршелла для вычисления матрицы лостижимости.

* 1. **Что делает алгоритм Уоршелла?**

Алгоритм Уоршелла используется для того, чтобы по матрице смежности ориентированного графа найти матрицу достижимости (или матрицу путей).

* + 1. **Матрица достижимости (T)**

Эта матрица показывает: можно ли попасть из вершины i в вершину j по одному или более рёбрам.

**Исходные данные**

Мы начинаем с матрицы смежности A, где:

A[i][j] = 1, если из вершины i есть ребро к j,

A[i][j] = 0, если нет ребра.

**Матрица смежности** — это способ записать граф в виде таблицы (или двумерного массива), где указано, есть ли ребро (связь) между двумя вершинами графа.

Это просто таблица из нулей и единиц, показывающая, как вершины графа соединены друг с другом.

**Основная идея алгоритма:**

Если из вершины i можно попасть в вершину k,

а из k можно попасть в j,

значит, из i можно попасть в j через k.  
 **Как работает алгоритм Уоршелла?**

Берём матрицу смежности A как исходную матрицу.

Копируем её в новую матрицу T — она станет матрицей достижимости.

Последовательно для каждой вершины k проверяем все пары вершин i и j.

Если из i можно попасть в k, и из k можно попасть в j,

то в T[i][j] записываем 1 — то есть из i можно попасть в j через k.

Повторяем шаг 4 для всех k, i, j.

**Матрица достижимости** — это таблица, которая показывает: можно ли добраться из одной вершины графа до другой, если пройти по одному или нескольким рёбрам (связям).

**Формула**:

T[i][j] = T[i][j] OR (T[i][k] AND T[k][j])  
  
**Пример**

A → B

A → C

B → C

C → A

D — изолирована

A B C D

-------------

A | 0 1 1 0

B | 0 0 1 0

C | 1 0 0 0

D | 0 0 0 0   
**Шаг 1:** через вершину A Мы проверяем:

можно ли попасть из i в j через A

Вершина C может попасть в B через A:C → A= 1 A → B= 1⇒ значит, C → B через A ⇒ 1Это новый путь, поэтому вписываем 1.

A B C D

-------------

A | 0 1 1 0

B | 0 0 1 0

C | 1 1 0 0 ← изменилось

D | 0 0 0 0

**Шаг 2**: через вершину B

Проверяем:

можно ли попасть через вершину B

Вершина B может попасть в A через C:

B → C: = 1 C → A: = 1 ⇒ путь B → C → A ⇒ ставим = 1

Вершина C может попасть в C через B:C → B: = 1 B → C: = 1 ⇒ цикл ⇒ = 1  
  
 A B C D

-------------

A | 0 1 1 0

B | 1 0 1 0 ← изменилось

C | 1 1 1 0 ← изменилось

D | 0 0 0 0  
  
**Шаг 3**: через вершину C

Проверяем:

можно ли пройти через C

A → C: уже есть

C → A: уже есть

A → A: A → C: 1

C → A: 1⇒ путь A → C → A ⇒ значит 1

B → B: B → C: = 1 C → B: 1⇒ B → C → B ⇒ цикл ⇒ 1

Всё остальное уже есть  
 A B C D

-------------

A | 1 1 1 0 ← изменилось

B | 1 1 1 0 ← изменилось

C | 1 1 1 0

D | 0 0 0 0

Шаг 4: через вершину D

D — изолированная вершина.

Ни в неё не входят, ни из неё не выходят — ничего не меняется.  
 A B C D

-------------

A | 1 1 1 0

B | 1 1 1 0

C | 1 1 1 0

D | 0 0 0 0