

¹ викладач, відокремлений структурний підрозділ Національного авіаційного університету, Слов'янський коледж НАУ

² викладач, відокремлений структурний підрозділ Національного авіаційного університету, Слов'янський коледж НАУ

e-mail: schensnevichov@gmail.com, yuraschensnevich@gmail.com

ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ФЛОЙДА-ВОРШЕЛЛА, ДЛЯ ПОШУКУ НАЙКОРОТШИХ ШЛЯХІВ МІЖ УСІМА ПАРАМИ ВЕРШИН ГРАФУ

В статті наведено реалізацію алгоритму Флойда-Воршелла для виводу матриці, яка містить довжини найкоротших шляхів між усіма вершинами графа, з використанням матриці суміжності та псевдокоду. Також виводить шукану максимальну найкоротшу відстань між двома вершинами засобами мови програмування C++.

Ключові слова: *граф, алгоритм, найкоротший шлях, алгоритм Флойда-Воршелла.*

Вступ

Найбільш вражаючим прикладом графа в сучасному світі є Інтернет. Саме інтернет з усім його різноманіттям і складністю форм є типовим графом. Його вузли - це адреси сторінок файлів, що знаходяться в мережі, а ребра — гіперпосилання, що зв'язують їх в місці. З іншого боку, комп'ютери, зв'язані між собою та утворюють всесвітню павутину, також можна розглядати як граф. Та інтернетом дуже складно управляти, оскільки передбачити всі можливі дзвінки і зв'язки просто ніхто не в силах. Для того щоб браузер на вашому комп'ютері міг завантажити сторінку, яка фізично знаходиться в Австралії, дані повинні пройти через кілька серверів, тобто у них буде існувати своєрідний маршрут у мережі.[2] Він буде автоматично вибиратися серверами таким чином, щоб його довжина була мінімальною (з точки зору графа), а дані по можливості проходили через незавантажені сегменти мережі. Через те, що навантаження в різних ділянках мережі постійно та випадковим чином змінюється, навіть якщо ви завантажите з одного й того ж сайту сторінку два рази поспіль з інтервалом в декілька секунд, вона може прийти до вас двома різними шляхами. Якщо бути більш точними та згадати про те, що в

мережі існують не сторінки в готовому вигляді, а пакети з даними, і навіть одна сторінка може складатися з декількох пакетів, то стає ясно: навіть фрагменти однієї сторінки можуть приходити в Інтернеті з одного комп'ютера на інший різними шляхами. Завдання доставки пакетів даних з одного вузла мережі (хоста) в інший називається завданням маршрутизації, і вона аж ніяк не є тривіальною.

Основна частина

Алгоритм Флойда-Воршелла – динамічний алгоритм для знаходження найкоротших відстаней між усіма вершинами зваженого орієнтованого графа. Розроблено в 1962 році Робертом Флойдом і Стівеном Воршеллом. При цьому алгоритм вперше розробив і опублікував Бернард Рой в 1959 році, але ця публікація пройшла повз увагу.[3]

Алгоритм

Нехай вершини графа $G = (V, E)$, $|V| = n$ пронумеровані від 1 до n і введено позначення d_{ij}^k для довжини найкоротшого шляху від i до j , який окрім самих вершин i, j проходить тільки через вершини $1 \dots k$. Очевидно, що d_{ij}^0 – довжина (вага) ребра (i, j) , якщо воно існує (в іншому разі його довжина може бути позначена як ∞). [5]

Існує два варіанти значення d_{ij}^k , $k \in 1, \dots, n$:

1. Найкоротший шлях між i, j не проходить через вершину k , тоді $d_{ij}^k = d_{ij}^{k-1}$.
2. Існує більш короткий шлях між i, j , що проходить через k , тоді він спочатку йде від i до k , а потім від k до j . У цьому випадку, очевидно, $d_{ij}^k = d_{ik}^{k-1} + d_{kj}^{k-1}$.

Таким чином, для знаходження значення функції досить вибрати мінімум з двох позначених значень.

Тоді рекурентна формула для d_{ij}^k має вигляд: d_{ij}^k – довжина ребра $d_{ij}^k = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1})$. Алгоритм Флойда-Воршелла послідовно обчислює всі значення d_{ij}^k , $\forall i, j$ для k від 1 до n . Отримані значення d_{ij}^n є довжинами найкоротших шляхів між вершинами i, j . [4]

Псевдокод

На кожному кроці алгоритм генерує матрицю W , $w_{ij} = d_{ij}^n$. $W_{ij} = d_{ij}^n$. Матриця W містить довжини найкоротших шляхів між усіма вершинами графа. Перед роботою алгоритму матриця W заповнюється довжинами ребер графа.[1]

```
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      W[i] [j] = min (W [i] [j], W [i] [k] + W [k] [j])
```

Складність алгоритму

Три вкладених цикли містять операцію, яка виконується за константний час. $\sum_{n,n,n} O(1) = O(n^3)$, тобто алгоритм має кубічну складність, при цьому простим розширенням можна отримати також інформацію про найкоротші шляхи, крім відстані між двома вузлами записувати в матрицю ідентифікатор першого вузла в дорозі.

Реалізація алгоритму Флойда-Воршелла мовою програмування C++.

Вхідними даними будуть кількість вершин графа n ($1 \leq n \leq 100$) та в наступних n рядках знаходиться по n чисел, які задають матрицю суміжності графа. Де -1 – це відсутність ребра між вершинами, а будь-яке невід’ємне число – присутність ребра даної ваги. Всі числа по модулю не перевищують 100. На головній діагоналі матриці завжди розташовані нулі.

У вихідних даних отримаємо матрицю найкоротших відстаней між парами вершин, тобто n рядків по n чисел. j -е число в i -ому рядку повинно дорівнювати вазі найкоротшого шляху з вершини i у вершину j . Виводиться шукане максимальна найкоротша відстань.

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(){
  int n;
  cout<<"n=";
  cin >> n;
  cout<<"Input matrix A \n";
  int a[n][n];
  long long maxr = 0;
  for( int i=0 ; i<n ; i++ ){
    for( int j=0 ; j<n ; j++ ){
      cin >> a[i][j];
    }
  }
}
```

```
for( int k=0 ; k<n ; k++ ){
    for( int i=0 ; i<n ; i++ ){
        for( int j=0 ; j<n ; j++ ){
            if(i != j && a[i][k] != -1 && a[k][j] != -1){
                if(a[i][j] == -1){
                    a[i][j] = a[i][k] + a[k][j];
                }
                else
                {
                    a[i][j] = min(a[i][j], a[i][k] + a[k][j]);
                }
            }
        }
    }
}

cout<< "matrix \n" ;
for( int i=0 ; i<n ; i++ ){
    for( int j=0 ; j<n ; j++ ){
        cout << a[i][j] << " ";
    }
    cout<< endl;
}

for( int i=0 ; i<n ; i++ )
{
    for( int j=0 ; j<n ; j++ )
    {
        if(a[i][j] > maxr) maxr = a[i][j];
    }
}

cout << "max rastoyanie = " <<maxr << endl;
}
```

Висновки

Алгоритм відшукування найкоротших шляхів Флойда-Воршелла використовують в мережах з централізованим управлінням інформаційними потоками, з метою автоматизації процесу та обліку трафіку. В реаліях сьогодення ми шукаємо найкоротший шлях для вирішення задач. Обираючи правильний алгоритм, можна скоротити витрати ресурсів та заощадити час.

Література

1. *Бондаренко М.Ф.* Комп'ютерна дискретна математика. / М.Ф. Бондаренко, Н.В. Білоус — Харків. «Компанія Сміт», 2004. — 480 с.
2. *Вишне夫斯基 В.М.* Теоретические основы проектирования компьютерных сетей. / В.М. Вишне夫斯基 — М.: Техносфера, 2003. — 512 с.
3. *Кормен Томас Х.* Алгоритмы: построение и анализ. / Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн., — М.: «Вильямс», 2006. — 1296 с.
4. *Левитин А.В.* Алгоритмы. Введение в разработку и анализ. / А.В. Левитин. — М.: Вильямс, 2006. — 576 с.
5. *Нікольський Ю.В.* Дискретна математика. / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина, — Київ. Видавнича група ВНУ, 2007. — 368 с.

Shchensnevych Olga V., Shchensnevych Yurii Y.

Slavyansk College of National aviation university, Sloviansk, Ukraine.

Program implementation of the Floyd-Warshall algorithm, for searching the shortest paths between all pairs of the vertices of the graph

The article presents the implementation of the Floyd-Warshall algorithm for outputting a matrix, which contains the lengths of the shortest paths between all vertices of the graph, using the adjacency matrix and pseudo-code. It also displays the desired maximum shortest distance between the two vertices using the C ++ programming language.

Keywords: *graph, algorithm, shortest path, Floyd-Warshall algorithm.*