

4. Гасратова Н. А., Столбовая М. В., Бойцов Д. С., Степанова Д. С. Математическая модель хищник-жертва на линейном ареале // Молодой ученый. – 2014.- №11. С. 1-10. — URL <https://moluch.ru/archive/70/12111/>

УДК 519.688

ПОШУК ОПТИМАЛЬНОЇ ВІДСТАНІ МІЖ ВЕРШИНАМИ ГРАФА ЗА АЛГОРИТМОМ ДЕЙКСТРИ ТА РЕАЛІЗАЦІЯ ЙОГО НА C ++

Онищенко Г.О.

*Таврійський державний агротехнологічний університет,
м. Мелітополь, Україна*

Summary: the paper considers the implementation of the algorithm for finding the optimal distance between the vertices of the graph using the Dijkstra algorithm in the C ++ programming language.

Keywords: dijkstra algorithm, graph, graph vertices, C ++ programming.

Сьогодні «Теорія графів» є одним із розділів дискретної математики, який стоїть поряд із математичним моделюванням та інтенсивно розвивається. В першу чергу, це пов'язано з широким використанням комп'ютера як засобу вирішення наукових і прикладних задач.

При вивченні дисциплін «Дискретна математика» у розділі теорії графів розглядаються такі теми: «Основні поняття теорії графів», «Типи графів», «Розфарбування графів», «Знаходження найменшого шляху», тощо.

Необхідно зазначити, що більшість задач цього розділу мають «цікаве» формулювання, задачі на графах дозволяють активно використовувати графічне зображення для пошуку розв'язку.

Графічне представлення можна отримати як на папері, так і з допомогою комп'ютерних програм обробки графів. Це в значній мірі розширює коло зацікавленості студента у вивченні матеріалу. Програми дозволяють легко редагувати зображення графа, що дає можливість вивчати і виявляти певні властивості різних класів графів, формулювати прості алгоритми рішення.

Нами була обрана задача пошуку оптимальної відстані за алгоритмом Дейкстри, яка є однією з найбільш потрібних на практиці в сучасній теорії графів.

Алгоритм і досі широко застосовується протоколами маршрутизації OSPF та IS-IS, в сфері сервісу, для виявлення послідовності доріг, які краще використовувати для транспортування товарів.

Мета роботи: реалізувати алгоритм пошуку оптимальної відстані за алгоритмом Дейкстри мовою програмування C++.

Маємо зважений орієнтований граф $G(V, E)$ без петель та дуг від'ємної ваги. Необхідно знайти найкоротші шляхи від деякої вершини a графа G до всіх інших вершин цього графа.

Сформулюємо алгоритм Дейкстри. Введемо наступні позначення:

- n – число вершин в графі;
- $r[i][j]$ – масив довжин ребер від вершини i до вершини j .

Від’ємне значення в комірці позначає відсутність ребра.

- $len[i]$ – шуканий масив (мінімальна відстань від вершини a до вершини i). Заздалегідь заповнений нескінченностями.

Для вирішення задач введемо додаткові змінні.

- $used[i]$ – масив прапорів. Заздалегідь заповнений $false$;
- k – поточна найближча до a вершина.

Алгоритм на псевдокодї формулюється наступним чином:

- 1 обрати в якості поточної вершини k вершину a
- 2 заповнити $len[a] = 0$
- 3 повторити $n-1$ разів
- 4 помітити вершину k як пройдену $used[k] = true$
- 5 для всіх не пройдених вершин i перевірити
- 6 якщо відстань $len[i] > len[k] + r[k][i]$
- 7 то змінити $len[i] = len[k] + r[k][i]$
- 8 знайти непройдену вершину з мінімальною відстанню $len[i]$ та

запам’ятати її номер в k

Приведений алгоритм реалізується наступним чином на мові C++:

```
// Ініціалізація матриці зв'язків
for (int i = 0; i < SIZE; i++) {
    a[i][i] = 0;
for (int j = i + 1; j < SIZE; j++) {
    printf("Введіть відстань %d - %d: ", i + 1, j + 1);
    scanf("%d", &temp);
    a[i][j] = temp;
    a[j][i] = temp; } }
//Ініціалізація вершин та відстаней
for (int i = 0; i < SIZE; i++) {
    d[i] = 10000; v[i] = 1; }
d[0] = 0; // Шаг алгоритму
do {
    minindex = 10000;
    min = 10000;
for (int i = 0; i < SIZE; i++)
    { // Якщо вершину ще не обійшли і вага менше min
if ((v[i] == 1) && (d[i] < min)) { // Перепишемо значення
min = d[i];
minindex = i; } }
// Додаємо знайдену мінімальну вагу до поточної ваги вершини та порівнюємо з
поточною мінімальною вагою вершини
if (minindex != 10000) {
for (int i = 0; i < SIZE; i++) {
if (a[minindex][i] > 0) {
temp = min + a[minindex][i];
if (temp < d[i]) {
d[i] = temp; } } } }
v[minindex] = 0; }
```

Додаємо знайдену мінімальну вагу до поточної ваги вершини та порівнюємо з поточною мінімальною вагою вершини

```
if (minindex != 10000) {
for (int i = 0; i < SIZE; i++) {
if (a[minindex][i] > 0) {
temp = min + a[minindex][i];
if (temp < d[i]) {
d[i] = temp; } } } }
v[minindex] = 0; }
```

```

} while (minindex < 10000); // Відновлення шляху
while (end > 0) // поки не дійшли до початкової вершини {
for(int i=0; i<SIZE; i++) // переглядаємо всі вершини
if (a[end][i] != 0) // якщо зв'язок є {
int temp = weight - a[end][i]; // знаходимо вагу шляху з попередньої вершини
if (temp == d[i]) // якщо вага співпала з розрахунковою
{ // значить з цієї вершини і був перехід
weight = temp; // зберігаємо нову вагу
end = i; // зберігаємо попередню вершину
ver[k] = i + 1; // і записуємо її в масив
k++; }
}
}

```

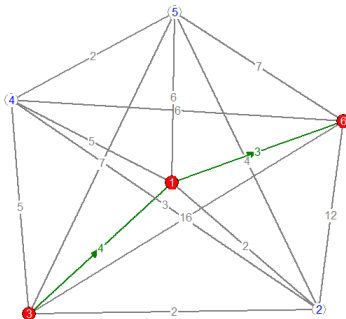


Рис. 1. Приклад заданого графу

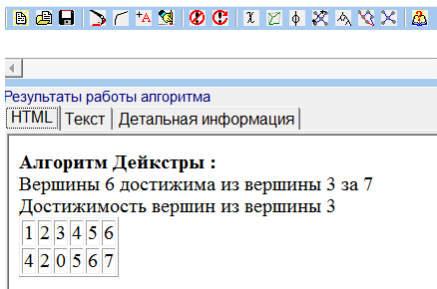


Рис. 2. Результат виконання алгоритму на прикладі графу (рис.1)

Висновки. Отже, нами було розглянуто алгоритм Дейкстри для пошуку найменших шляхів між заданими точками та реалізовано його мовою програмування C++. Дана програма допоможе візуалізувати результат алгоритму, спростити подальші розрахунки та заохотити до вивчення даної теми більшу кількість студентів.

Список літератури

1. Капітонова Ю.В., Кривий С.Л., Летичевський О.А., Луцький Г.М., Печурін М.К. Основи дискретної математики. К.: Наукова думка, 2002. – 580 с.
2. Свами М., Тхуласираман К. Графы, сети, алгоритмы. – М.: Мир, 1984. – 455 с.

3. Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике. – М.: Наука, 1977. – 368 с.

4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. – 536 с.

УДК 004.72

СЕМАНТИЧНЕ ОПРАЦЮВАННЯ РЕЗУЛЬТАТІВ ФОРМАЛЬНОГО Й ІНФОРМАЛЬНОГО НАВЧАННЯ

Прийма С.М., д.п.н.

Таврійський державний агротехнологічний університет,
м. Мелітополь, Україна

***Summary:** the paper analyzes publications related to the development trends of national qualifications systems, which should link the market of educational services to the labor market. Such an analysis suggests that an effective tool for solving this problem is ESCO – the Multilingual Classification of European Skills, Competences, Qualifications and Professionals. ESCO defines and classifies skills (both "soft" and "hard"), competences, qualifications and occupations that are relevant to the European labor market, education and training.*

***Keywords:** on-formal and informal learning, labour market, market of educational services, occupation, knowledge, skill, ESCO.*

Постановка проблеми. Євроінтеграційні процеси в Україні, збалансування національних інтересів на ринку освітніх послуг і ринку праці, підвищення якості підготовки кадрів, з одного боку, і глобалізація, технологічні, економічні й демографічні процеси, зростаюча мобільність трудових ресурсів – з іншого, потребують розроблення і операціоналізації національної і регіональної політики в галузі кваліфікацій, інституційних механізмів і процесів забезпечення якості, оцінки та присудження кваліфікацій, визнання результатів навчання (як формального, так і неформального й інформального) - знань, умінь та компетентностей, що пов'язують ринок освітніх послуг з ринком праці. Успішне функціонування такої політики сприятиме розвитку як вертикальної мобільності громадян в просторі професійної освіти, так і горизонтальній мобільності з однієї професійної траєкторії на іншу за рахунок механізму офіційного визнання часткових кваліфікацій, набутих в системі формального, неформального й інформального навчання. Як зазначено в Рекомендаціях Ради Європи про визнання неформального й інформального навчання, «... визнання результатів навчання (знань, умінь і навичок та компетенцій), досягнутих через неформальне й інформальне навчання, в тому числі через відкриті освітні ресурси, є необхідним для доступу громадян до ринку праці та безперервного навчання».