для кожної координати розв'язання. Таким чином, кожна координата розв'язку оцінюється вектором $k^j, j = 1, \ldots, n$, а координати розв'язку - $k^j, j = 1, \ldots, n$, а координати оцінюються значеннями критеріїв, які впорядковані за ступенем важливості. Вектор $k^j, j = 1, \ldots, n$ у лексикографічному спуску називається вектором $k^j, j = 1, \ldots, n$, n у лексикографічному порядку спадання призведе до нового порядку або масиву змінних, в якому алгоритм лексикографічного пошуку буде запущене заново.

Наявні алгоритми лексичного пошуку призначені для пошуку найкращого, або близького до оптимального рішення в одному порядку. Алгоритми лексичного пошуку є важливими інструментами для вирішення проблем, пов'язаних із пошуком і порівнянням об'єктів відповідно до лексичного порядку. Ці алгоритми забезпечують швидкий і ефективний пошук і відіграють важливу роль у багатьох галузях, як-от оброблення текстів, бази даних і криптографія.

Лексичні алгоритми пошуку можна використовувати для пошуку рішень, проблем шляхом порівняння можливих альтернатив і вибору найкращого рішення. Ці алгоритми можна використовувати в багатьох галузях, де потрібно шукати і порівнювати різні об'єкти, в тому числі в сферах, пов'язаних з наукою, технікою і економікою.

Таким чином, алгоритми лексичного пошуку є важливим інструментом у багатьох галузях, і їх застосування дозволяє ефективно і швидко знаходити рішення та альтернативи в різних сферах діяльності.

Список літератури

1. Сергієнко І.В. Завдання дискретної оптимізації. Проблеми, методи вирішення, дослідження/І.В. Сергієнко, В.П. Шило. - К.: Наукова думка, 2003. - 264 с.

2. Чупов С.В. Нові підходи до вирішення завдань дискретного програмування на основі лексикографічного пошуку / С.В.Чупов // Кібернетика та систем. аналіз. – 2016. – № 4.– С. 43-54.

УДК 00442.519.1

Діброва І. С., студент 1 курсу
спеціальності 122 «Комп'ютерні науки»
Ніколюк П.К., д.ф.-м.н., професор, професор
кафедри інформаційних технологій

ЗАСТОСУВАННЯ АЛГОРИТМУ ДЕЙКСТРИ ДЛЯ ЕФЕКТИВНОГО ВИЗНАЧЕННЯ НАЙКОРОТШОГО ШЛЯХУ В ГРАФАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Новітні дослідження показують, що застосування алгоритму Дейкстри у розробці програмного забезпечення дозволяє зменшити час та зусилля, необхідні для визначення найкоротшого шляху в графах з великою кількістю вершин та ребер. Застосування цього алгоритму можливе в різних галузях, включаючи транспортну логістику, маршрутизацію мережі, планування маршрутів дронів та багато інших [1].

При розробці програмного забезпечення для використання алгоритму Дейкстри важливо враховувати такі фактори, як швидкість обчислення та зберігання даних, а також можливість використання різних форматів даних, які можуть бути представлені у вигляді графів [2].

Отже, застосування алгоритму Дейкстри та розроблення програмного забезпечення для використання цього алгоритму є дуже важливими та актуальними завданнями в різних галузях, що можуть допомогти підвищити ефективність та зменшити витрати в цих галузях.

Розглянемо наступний граф.

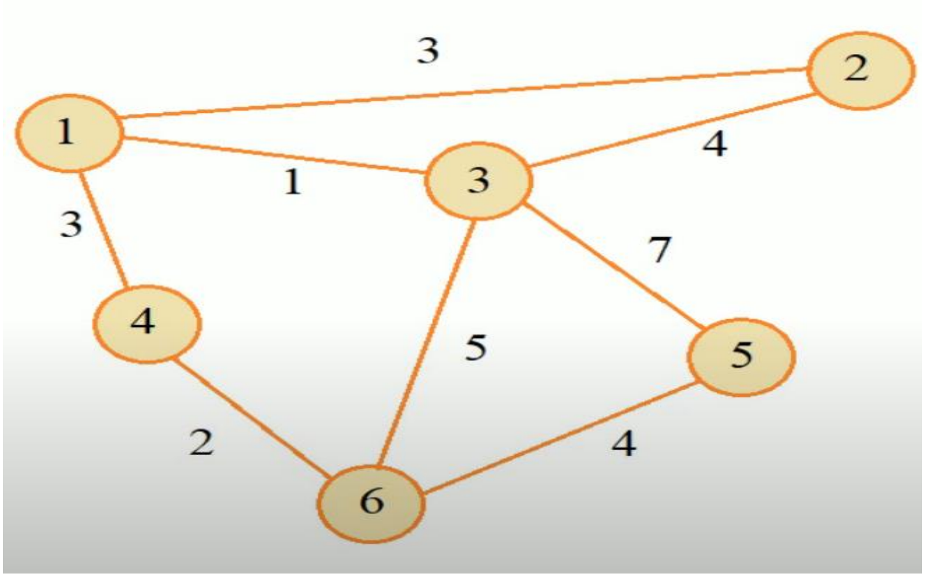

Рисунок 1 – представлення дорожнього маршруту у вигляді графа

Основна ідея алгоритму дуже проста. Фактично тут використовується повний направлений пошук. Спочатку треба визначитись із стартовою вершиною, нехай це буде вершина №1. Далі зручно використовувати наступну таблицю:

№	1	2	3	4	5	6
1	0	∞	∞	∞	∞	∞

Рисунок 2 – Таблиця для представлення алгоритму Дейкстри

Де по вертикалі відображені номера ітерації, а по горизонталі нумерація вершин. На першій ітерації для стартової вершини вписуємо 0, адже ця вершина є початкової і ми вже знаходимось в ній. В інших вершинах – безкінечність, бо ще не виначились, які мінімальні ваги знадобляться для переходу в інші вершини.

На другій ітерації розглядаємо всі вершини, в які можна прийти із початкової, визначаємо сумарні ваги та вибираємо найменше значення. Наступні інтерації розглядаємо за аналогічним приницпом та маємо:

№	1	2	3	4	5	6
1	0	∞	∞	∞	∞	∞
2		3	1	3	∞	∞
3		3		3	8	6
4				3	8	6
5					8	5
6					8	

Рисунок 3 – Представлення мінімальних ваг при переході з початкової вершини у всі інші

Розглянемо програмний варіант виконання цього алгоритму на мові програмування Python.

У першій частині коду визначається матриця суміжності графа “D”. Далі запускається цикл, який проходить по усіх вершинах графа і обчислює найкоротший шлях від початкової вершини до кожної з них.

```
D = ((0, 3, 1, 3, 0, 0),
     (3, 0, 4, 0, 0, 0),
     (1, 4, 0, 0, 7, 5),
     (3, 0, 0, 0, 0, 2),
     (0, 0, 7, 0, 0, 4),
     (0, 0, 5, 2, 4, 0))
```

Рисунок 4 – Матриця суміжності графа “D”

Кожен крок циклу складається з двох частин: перша частина перебирає всі вершини, які з’єднані з початковим ребром, та оновлює вагу найкоротшого шляху до кожної з них.

```
while v != -1:
    for j in get_link_v(v, D):
        if j not in S:
            w = T[v] + D[v][j]
            if w < T[j]:
                T[j] = w
```

У другій частині коду ми вибираємо наступну вершину, яка не була пройдена раніше і має найменший ваговий коефіцієнт серед усіх інших.

```
    v = arg_min(T, S)
    if v > 0:
        S.add(v)
```

Ця програма реалізує алгоритм Дейкстри для знаходження найкоротшого шляху від однієї вершини до всіх інших вершин. Програма виводить список T, де кожен елемент T[i] містить відстань від стартової вершини до i-ї вершини в графі.

```
[0, 3, 1, 3, 8, 5]

Process finished with exit code 0
```

Рисунок 5– результат роботи програми

Можемо зробити висновок, що програма працює коректно, адже дані таблиці на рисунку №3, які виділені зеленим кольором та дані роботи програми збігаються.

Ця програма може бути корисною для багатьох застосувань, наприклад, для побудови маршруту у GPS-навігації, для планування маршрутів логістичних компаній, для оптимізації маршруту руху роботів на виробництві та багатьох інших задач, де важливо знайти найкоротший шлях між двома точками в графі.

Список літературних джерел

1. *Cormen, T.H., Leiserson, C. E., Rivest, R.L., & Stein, C.(2009). Intoduction to algoritms. MIT press.*
2. *Nagendra, C.P., & Kumar, K. M. (2016). A study on shortest path algoritms in graph theory. International Journal of Computer Science and Mobile Computing, 5(3), 110-118.*
3. *Adorf, C.S. (2014). Dijkstra's shortest path algorithm. In C.S. Adorf & J.R. Wilson (Eds.), Encyclopedia of Operations Research and Management Science (3rd ed., pp. 410-413). Wiley.*

УДК 629.737.5:519.87

Діброва І. С., студент 1 курсу спеціальності 122 «Комп'ютерні науки»
Комаров В.Ф., к.т.н., завідувач навчально-практичної лабораторії інтелектуальних систем і мереж

ПРОГРАМУВАННЯ ТА НАЛАШТУВАННЯ ДРОНІВ З ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ АЛГОРИТМІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі застосування дронів зростає з кожним днем. Використання дронів є актуальним у різних галузях, включаючи логістику, агрокультуру, моніторинг навколишнього середовища та безпеку. Програмування та налаштування дронів з використанням математичних