Міністерство освіти і науки України
Дніпровський національний університет імені Олеся Гончара
Факультет прикладної математики і комп’ютерних технологій
Кафедра комп’ютерних технологій

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 3

з курсу «Моделі та методи штучного інтелекту»

на тему «Задача про туриста»

Варіант №14

Виконав:
студент гр. ПА-22-2
Овдієнко Андрій

Дніпро
2025

Зміст

1. Постановка задачі

Розв’язати задачу про туриста, використовуючи такий метод пошуку, як метод Дейскри.

2. Опис розв’язку

Туристична задача - це графова задача, потрібно знайти найкоротший шлях з точки А в точку Б. Класичні методи перебору неефективні з великою кількістю міст - саме тому існують алгоритми, такі як Дейскри. Цей алгоритм працює лише з невід’ємними ребрами.

Щоб зрозуміти, як це працює, уявіть, що у нас є набір різних міст, які розкидані в різних місцях, деякі з них з’єднані дорогами з іншими містами, і ми можемо використовувати дорогу, щоб дістатися з одного міста до іншого, час подорожі може бути різним, одна дорога займе 3 хвилини, інша - 4, і найважливіше питання - яка дорога найкоротша з точки А в точку Б.

Продемонструємо граф(Рисунок 1), який буде використовуватися у коді, зазначимо, що він буде неорітованим.

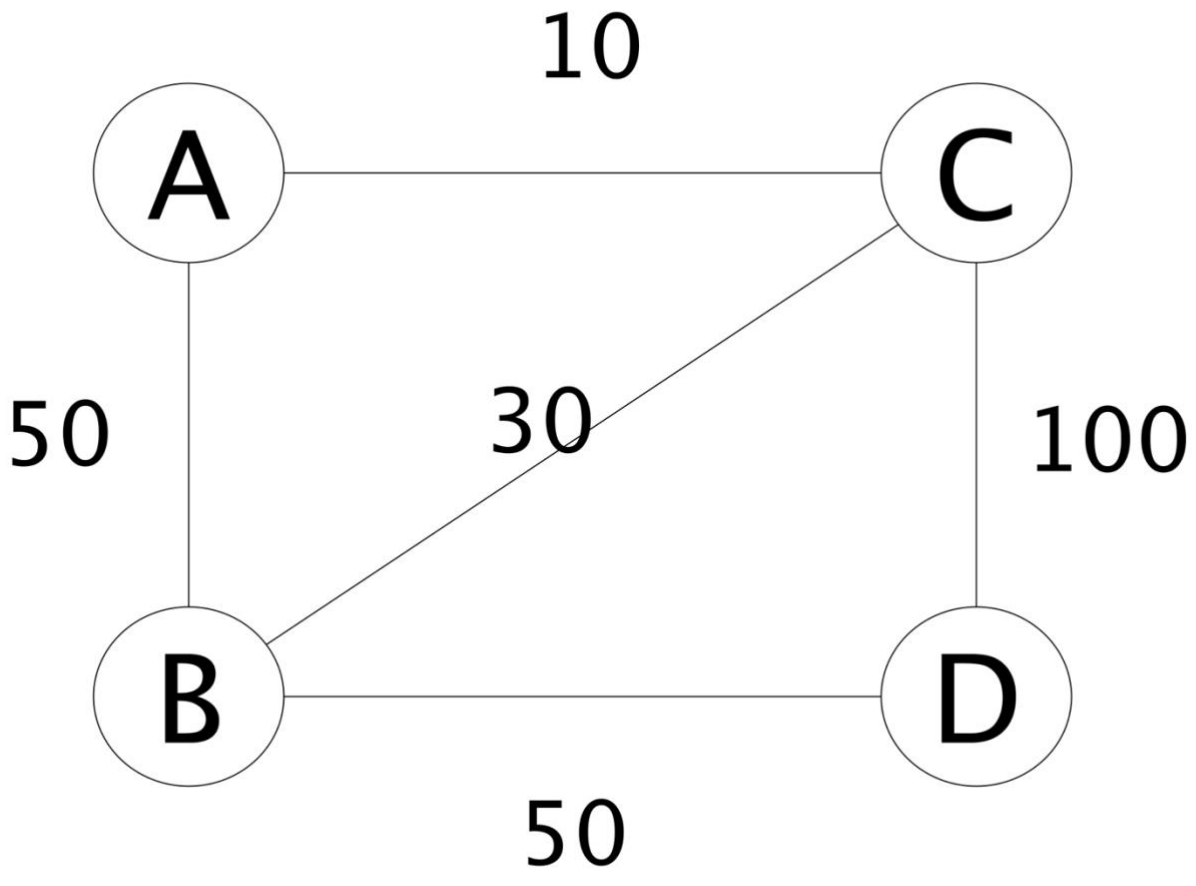

Рисунок 1 – Неорінтований граф.

Припустимо, нам потрібно дістатися з точки A до точки D. Ми використовуємо алгоритм Дейскре. Позначимо точку A значенням 0 - тому що ми починаємо звідти, а в усіх інших містах нескінченність(Рисунок 2).

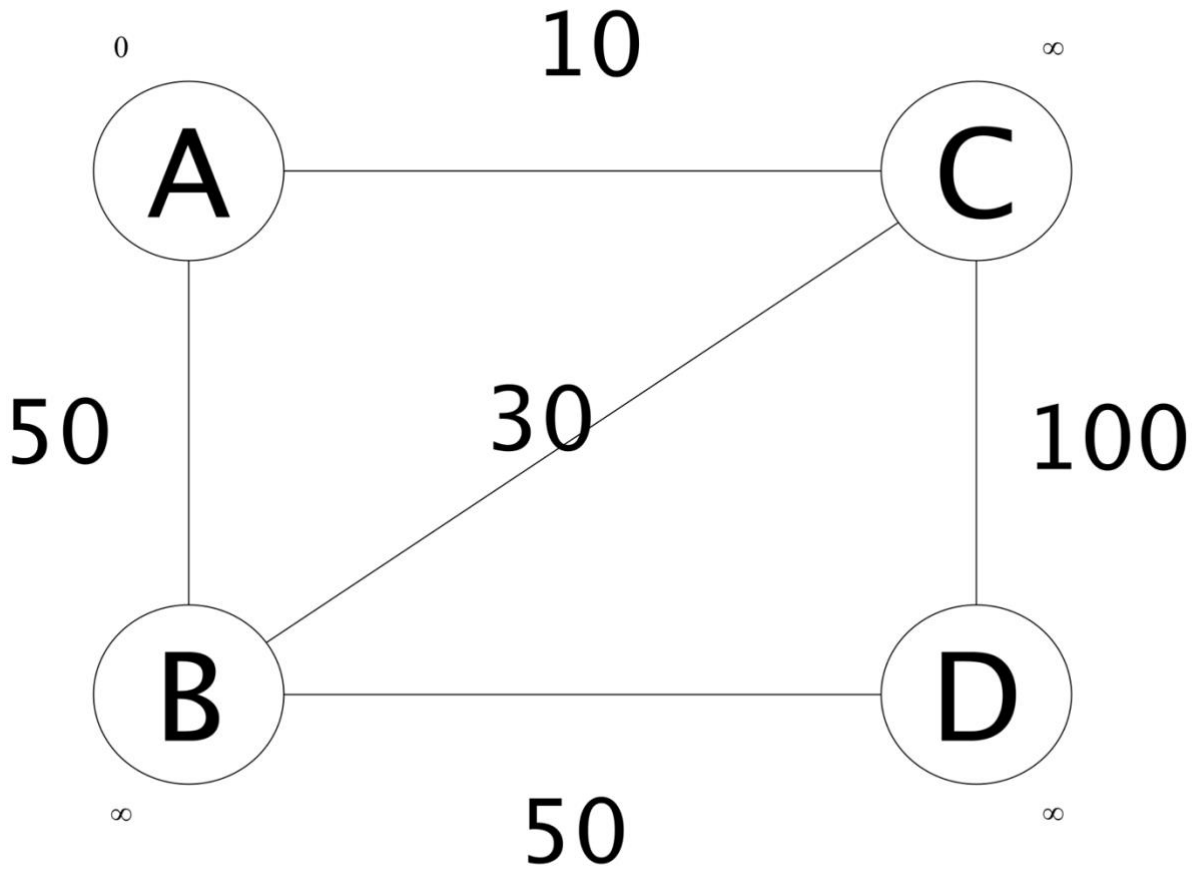

Рисунок 2 – Початкова оцінка.

Давайте розглянемо точку A та знайдемо, яке значення буде сусідніх точках, і порівняємо його зі значенням, яке вже існує.

Якщо ми порівняємо точки A та C, відстань дорівнює 10, поточне значення C дорівнює нескінченності, тому нова оцінка C дорівнює 10, оскільки 10 менше за нескінченність. Аналогічно, для B -50 менше за нескінченність, тому нова оцінка - 50.

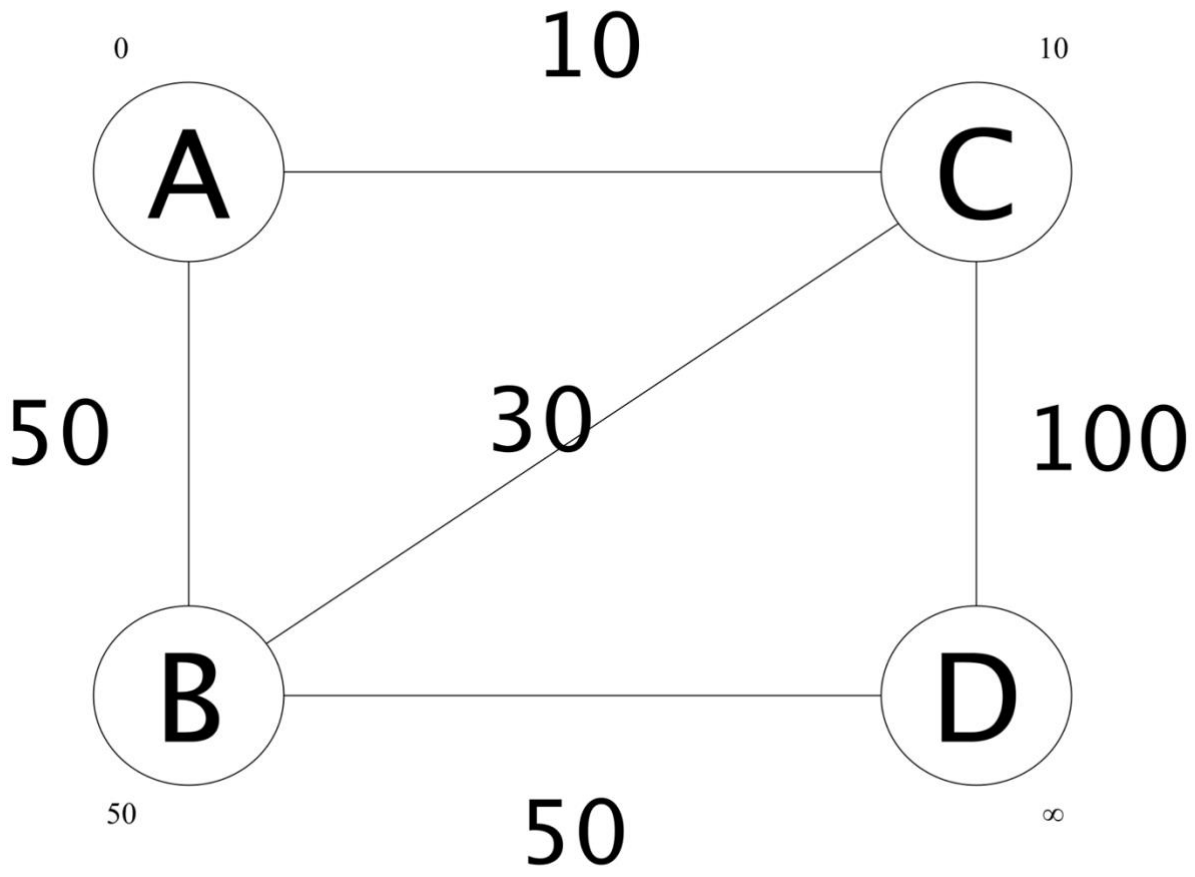

Рисунок 3 – Оцінка сусідів.

Тепер перейдемо до міста, яке має мінімальну оцінку і також не було проаналізовано - C. З C ми бачимо 3 шляхи: A, B, C – але нова оцінка вже не буде A – 10; B – 30; D – 100; а додасться поточна оцінка, тобто плюс 10. Отримаємо A – 20; B – 40; D – 110. B, D – стали меншими відповідно, тому їхній бал можна замінити, A – ні.

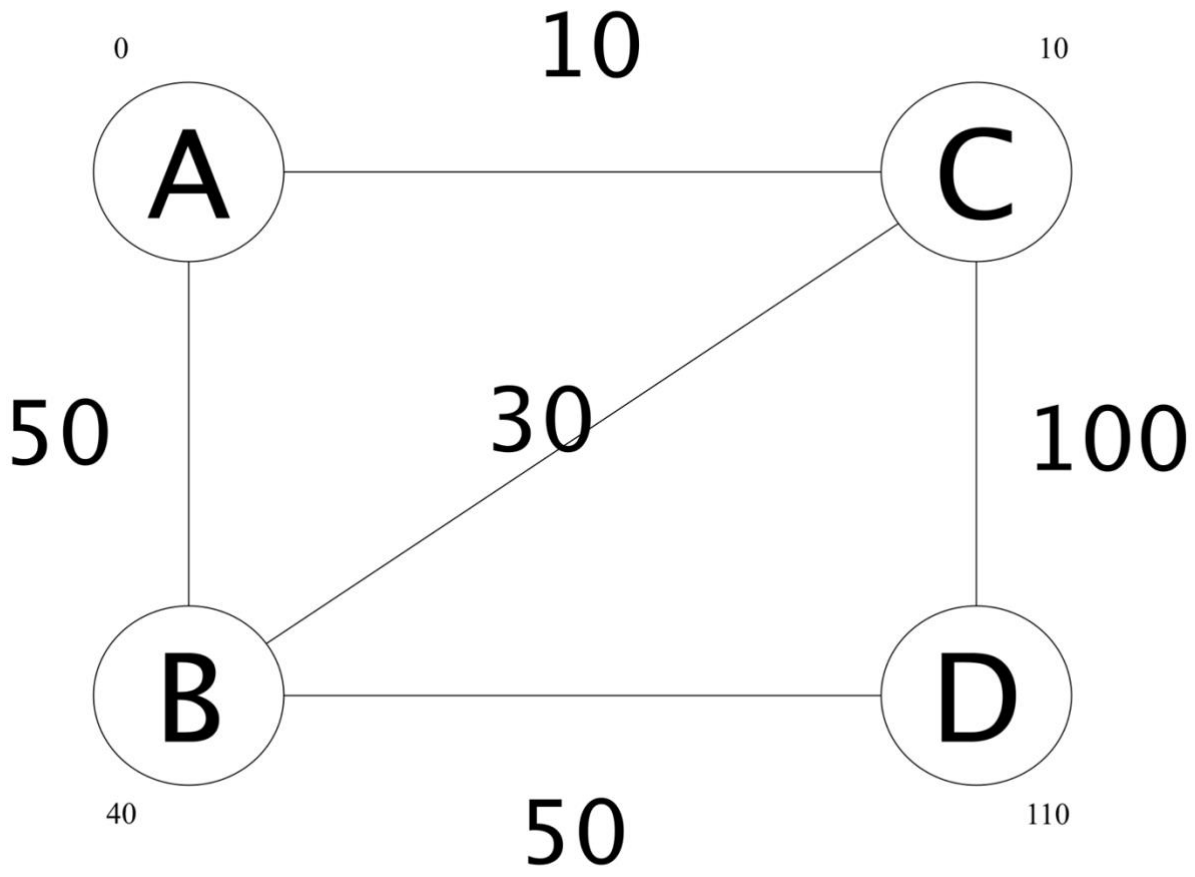

Рисунок 4 – Аналіз пункта C.

Бачимо, що таким чином вартість B змінилася з 50 до 40 – тобто відбулася релаксація. Далі, місто, яке не аналізувалося та має найнижчу вартість – B. Аналізуємо всіх сусідів, генеруємо шляхи: A-90; C-70; D-90; покращити можна лише D.

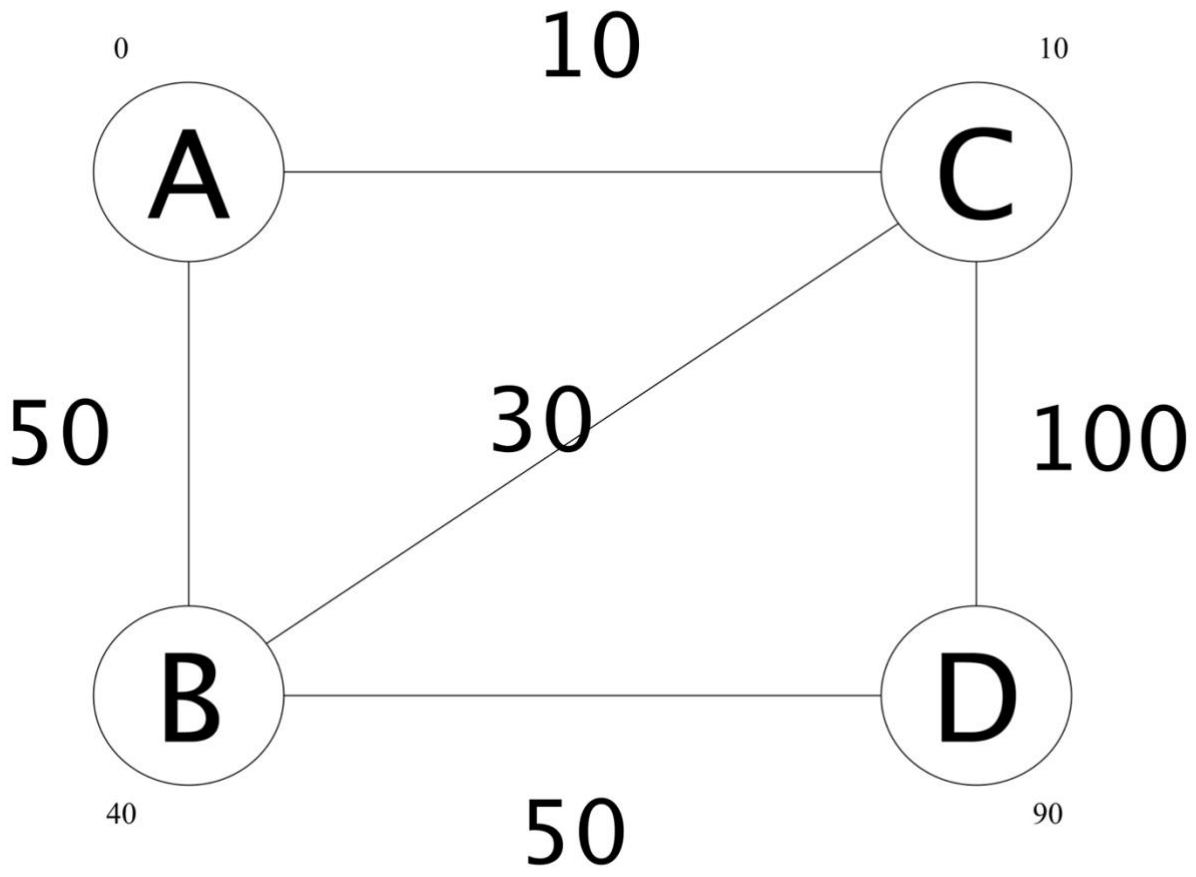

Рисунок 5 – Аналіз пункта B.

Залишається місто D, яке не аналізувалося та має найнижчу вартість. Тоді шляхи не можна покращити до B та C. Усі міста проаналізовані - тому ми маємо кінцеву відстань від точки A до точки D -90.

Щоб відтворити шлях, необхідно та достатньо оновити в пам'яті значення, з якого ми прийшли до цієї точки. Наприклад, у точці D зберігаємо значення B, оскільки звідти ми прийшли до D з мінімальними витратами.

Після довгого нерозуміння та перегляду відео:

“Алгоритм Дейкстры”, “Идея алгоритма Дейкстры”, “"Найкоротший шлях в графі" на Python, Haskell та на Prolog.”

– де демонструється робота SWI Prolog – було прийняте рішення перейти у Visual Studio та заванжити нове розширення для роботи з прологом. Доречі, в останньому відео хоч і написано Дейскри – але це повний перебір у глибину.

3. Вихідний текст програми розв’язку задачі

(основні фрагменти з коментарями)

```
:- dynamic city/4.

node(a,b,50).
node(b,a,50).

node(a,c,10).
node(c,a,10).

node(b,c,30).
node(c,b,30).

node(c,d,100).
node(d,c,100).

node(b,d,50).
node(d,b,50).

newcity:-
retractall(city(_,_,_,_)),
forall(node(X,Y,_),
(
((\+ (city(X,_,_,_))) -> assert(city(X,1000000000,0,none)) ; true),

((\+ (city(Y,_,_,_))) -> assert(city(Y,1000000000,0,none)) ; true)
)).

findway(A,B,Size,Last):-

newcity,
retract(city(A,_,_,_)),
assert(city(A,0,0,none)),
```

```
deskre,

city(B,Size,_,_),

path(B,Last).

path(B,Path):-
path(B,[],Path).

path(none, MustReturnResult, MustReturnResult):-!.

path(Element,LastPath,NewPath):-
city(Element,_,_,LastElement),
path(LastElement,[Element|LastPath],NewPath).

deskre:-
setof(Cost-Name, city(Name,Cost,0,_), Result),
(
Result=[] -> true ; (

Result = [_-NameBest|_],

cost(NameBest),
visited(NameBest),
deskre
)

).

deskre:-true.

cost(A):-
city(A,OldCost,_,_),
forall(node(A,B,TempCost),
```

```
(

NewCost is TempCost + OldCost,



city(B,Cost,0,_),

NewCost < Cost ->(
retract(city(B,_,C,_)),
assert(city(B,NewCost,C,A))
);true
)

).

visited(A):-
retract(city(A,B,_,D)),
assert(city(A,B,1,D)).
```

4. Опис інтерфейсу програми (керівництво користувача)

Щоб запустити програму – слід запустити термінал і перейти до відповідної папки з файлом main.pl – де було написано основний код. У командному рядку введіть: “swipl main.pl” – щоб завантажити файл main.pl. Далі – щоб виконати запит типу goal – потрібно його зареєструвати, в даному випадку – це команда: “findway(a,d,Size,Path).” – де a і d – це міста, між якими обчислюється відстань і шукається мінімум, Size – довжина, Path – повний шлях.

Під час налагодження команди типу listing(city). і listing(node). – були використані для перегляду фактів відповідних назв, але вам не обов’язково це знати.

5. Опис тестових прикладів

Для тестових прикладів нагадаємо наш граф.

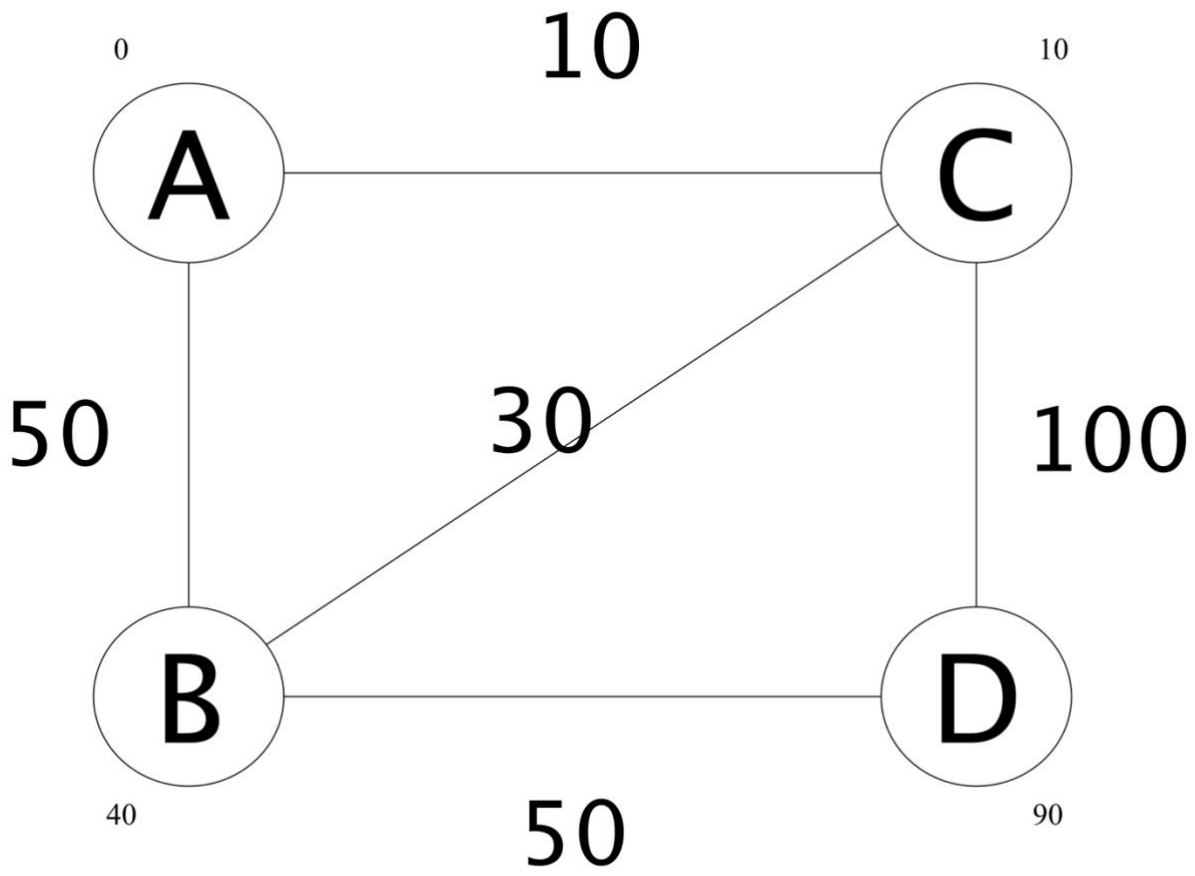

Рисунок 6 – Фінальна оцінка всіх шляхів від A до D.

5.1 Приклад 1

Дізнаємося фінальний шлях та оцінку від пункта A до D.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(a,d,Size,Path).
Size = 90,
Path = [a, c, b, d] .
```

Рисунок 7 – Результат роботи програми для мінімальної відстані від пункта A до пункта D. Відстань: 90. Шлях: [a, c, b, d].

5.2 Приклад 2

Дізнаємося фінальний шлях та оцінку від пункта D до A.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(d,a,Size,Path).
Size = 90,
Path = [d, b, c, a] 
```

Рисунок 8 – Результат роботи програми для мінімальної відстані від пункта D до пункта A. Відстань: 90. Шлях: [d, b, c, a].

5.3 Приклад 3

Дізнаємося фінальний шлях та оцінку від пункта A до B.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(a,b,Size,Path).
Size = 40,
Path = [a, c, b] 
```

Рисунок 9 – Результат роботи програми для мінімальної відстані від пункта A до пункта B. Відстань: 40. Шлях: [a, c, b].

5.4 Приклад 4

Дізнаємося фінальний шлях та оцінку від пункта B до A.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-  findway(b,a,Size,Path).
Size = 40,
Path = [b, c, a] 
```

Рисунок 10 – Результат роботи програми для мінімальної відстані від пункта B до пункта A. Відстань: 40. Шлях: [b, c, a].

5.5 Приклад 5

Дізнаємося фінальний шлях та оцінку від пункта A до A.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(a,a,Size,Path).
Size = 0,
Path = [a] .
```

Рисунок 11 – Результат роботи програми для мінімальної відстані від пункта A до пункта A. Відстань: 0. Шлях: [a].

5.6 Приклад 6

Дізнаємося фінальний шлях та оцінку від пункта A до C.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(a,c,Size,Path).
Size = 10,
Path = [a, c] 
```

Рисунок 12 – Результат роботи програми для мінімальної відстані від пункта A до пункта C. Відстань: 10. Шлях: [a, c].

5.7 Приклад 7

Дізнаємося фінальний шлях та оцінку від пункта C до A.

```
/bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Labs/Lab_3/run.bat"
admin@MacBook-Pro-admin Lab_3 % /bin/bash "/Users/admin/Documents/My documends/University/DNU/4 cours/ИИ/Lab
s/Lab_3/run.bat"
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- findway(c,a,Size,Path).
Size = 10,
Path = [c, a] 
```

Рисунок 13 – Результат роботи програми для мінімальної відстані від пункта C до пункта A. Відстань: 10. Шлях: [c, a].

5.8 Приклад 8

Дізнаємося фінальний шлях та оцінку від пункта B до D.

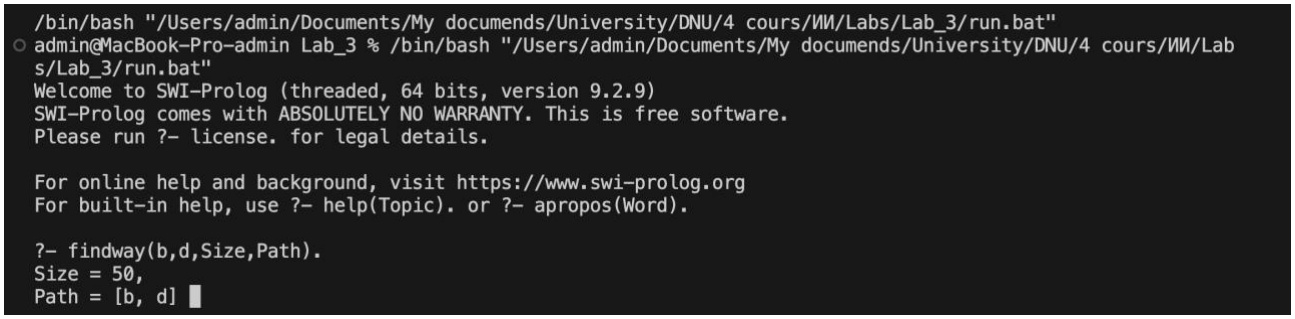

Рисунок 14 – Результат роботи програми для мінімальної відстані від пункта B до пункта D.

Відстань: 50. Шлях: [b, d].

6. Висновки

Дізналися, що таке задача про туриста. Створили неорієнтований граф. Освоїли оновлену мову Prolog (SWI-Prolog). Вивчили та засвоїли алгоритм Дейскри, який шукає найкорошу відстань між двома містами. Реалізували даний алгоритм на мові SWI-Prolog.