Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій

Кафедра САП

Методичні вказівки до лабораторної роботи №2.

з дисципліни: “Системи штучного інтелекту”

на тему:

“***Інформативні методи пошуку рішення задач в просторі станів. Алгоритм А*”.***

Затверджено на засіданні кафедри
“Системи автоматизованого проектування”
Протокол № 1 від 28.08.2025 р.

Львів-2025р.

ІНФОРМАТИВНІ МЕТОДИ ПОШУКУ РІШЕННЯ ЗАДАЧ В ПРОСТОРІ СТАНІВ. АЛГОРИМ А*. Методичні вказівки до лабораторної роботи №2 з курсу “Системи штучного інтелекту” для студентів базового напрямку “Комп’ютерні науки” / Укл. Р. І. Головацький. – Львів: НУЛП, 2025. – 21 с.

Укладач: Р. І. Головацький

1. МЕТА РОБОТИ

Мета роботи - Ознайомитись з інформативними методами пошуку рішення задач в просторі станів, типовим представником яких є алгоритм А*. Навчитись застосовувати алгоритм А* на практиці.

2. ТЕОРЕТИЧНІ ВІДОМОСТІ

Уявімо собі, що нам необхідно потрапити з пункту А в пункт В. Прямий шлях між цими двома пунктами розділений стіною, як показано на рисунку 1.

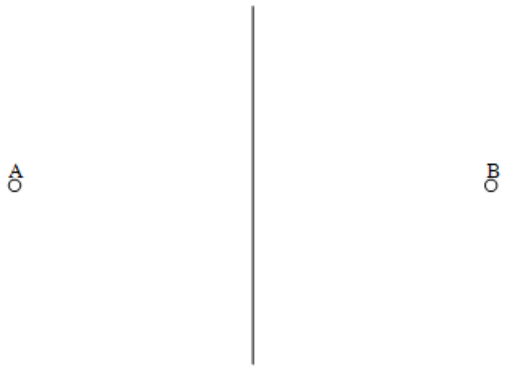


Рис. 1. Прямий шлях між пунктами А та В.

Для спрощення області пошуку шляху, її необхідно розділити на сітку з квадратними комірками (рисунок 2). Це дозволить визначити область пошуку шляху через двовимірний масив, кожен елемент якого буде представляти одну з кліток отриманої сітки. Одним із значень кожної комірки масиву буде прохідність відповідної їй клітинки (прохідна або непрохідна).

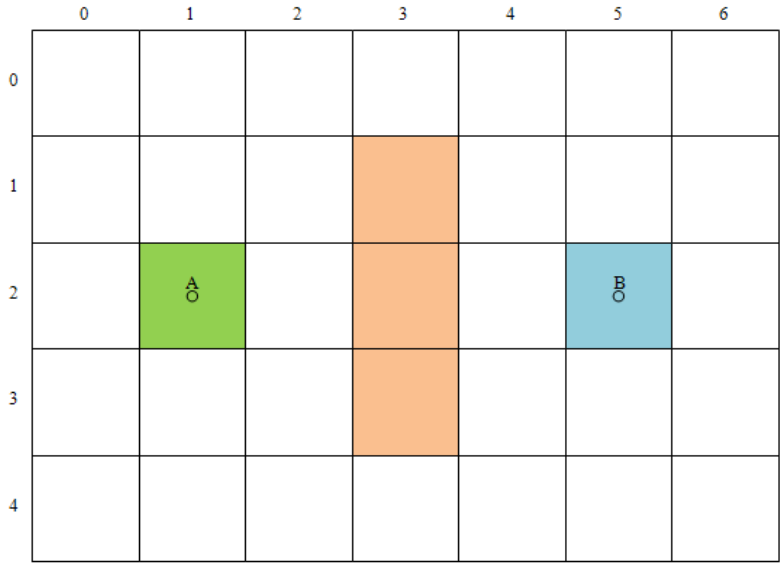


Рис. 2. Сітка з квадратними комірками.

Для знаходження шляху необхідно визначити, які саме клітини потрібні для переміщення з пункту А в пункт В. Як тільки шлях буде знайдений, можна

починати переміщення з центру клітини А в центр наступної клітини шляху, до тих пір, поки не буде досягнута кінцева точка В.

***Зауваження**: Сітка не обов'язково будується з квадратних комірок. Вона може бути побудована з прямокутних комірок, з шестикутних комірок, з трикутних комірок, або з будь-яких інших комірок. При цьому центральні точки комірок називають вершинами. Вершини можуть розташовуватися як в центрі, так і вздовж меж, або ще де-небудь.*

Пошук шляху починаємо зі наступних кроків:

1. Додаємо стартову клітку, де знаходиться точка А, в «відкритий список». В даний момент в цьому списку буде знаходитися тільки одна комірка, але пізніше в нього будуть додаватися й інші комірки. Клітини, що знаходяться у відкритому списку це клітини, які необхідно перевірити і вирішити, чи будуть вони частиною шуканого шляху до кінцевого пункту.
2. Шукаємо прохідні клітини, які межують зі стартовою кліткою, ігноруючи непрохідні клітини (зі стінами, водою та іншим), і додаємо їх у відкритий список. Для кожної з цих клітин зберігаємо клітку А як «батьківську».
3. Видаляємо стартову клітку А з відкритого списку і додаємо її в «закритий список» клітин, які більше не потрібно перевіряти.

Після цих кроків має вийти щось схоже на те, що зображено на рисунку 3. На ньому стартова клітина виділена оранжевим кольором для відображення того, що вона знаходиться в закритому списку. Всі сусідні клітини в даний момент знаходяться у відкритому списку, - вони виділені синім кольором. Кожна з цих клітин має вказівник, спрямований на батьківську клітку, яка в даному випадку є стартовою кліткою А.

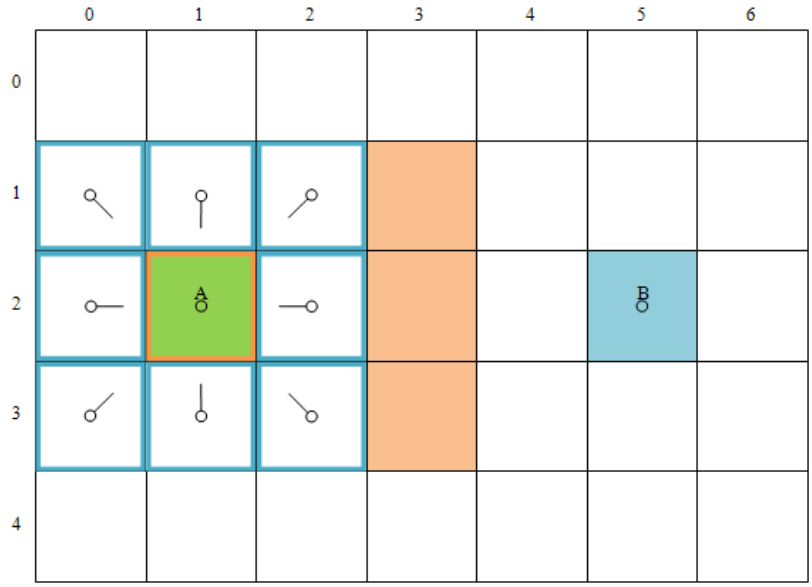


Рис. 3. Зображення клітин, що знаходяться у закритому та відкритому списках.

Далі необхідно вибрати одну клітку, що знаходяться у відкритому списку клітин, і, практично, повторити вищеописаний процес. Але наступна клітина з відкритого списку, вибирається не випадково, а в залежності від її величини F.

Величина F обчислюється за формулою 1:

$$F = G + H \quad (1)$$

де

- G - енергія, що витрачається на пересування із стартовою клітини A в поточну розглянуту клітку, слідуючи знайденому шляху до цієї клітинки;
- H - приблизну кількість енергії, що витрачається на пересування від поточної клітинки до цільової клітинки B. Спочатку ця величина дорівнює орієнтовному значенню, такому, що якби ми йшли безпосередньо, ігноруючи перешкоди (але виключивши діагональні переміщення). В процесі пошуку вона коригується в залежності від перешкод, що зустрічаються на шляху.

Зазвичай, енергія, що витрачається на проходження в сусідню клітку по горизонталі, береться рівною 10 одиницям, а по діагоналі - 14 одиницям. Для обчислення величини G поточної розглянутої клітини, необхідно величину G її батьківської клітини скласти з 10 або 14 (в залежності від діагонального або ортогонального розташування поточної клітини щодо батьківської клітини). Величина H зазвичай обчислюється методом Манхеттена (Відстань міських кварталів). Суть його полягає в тому, щоб порахувати загальну кількість клітин, необхідних для досягнення цільової клітини B, від поточної розглянутої клітини, причому ігноруючи діагональні переміщення між клітинами, а також будь-які перешкоди. Потім отриману кількість множиться на 10. Розрахувавши величину F за формулою 1, для всіх клітин у відкритому списку, отримаємо результат схожий на той, що зображено на рисунку 4.

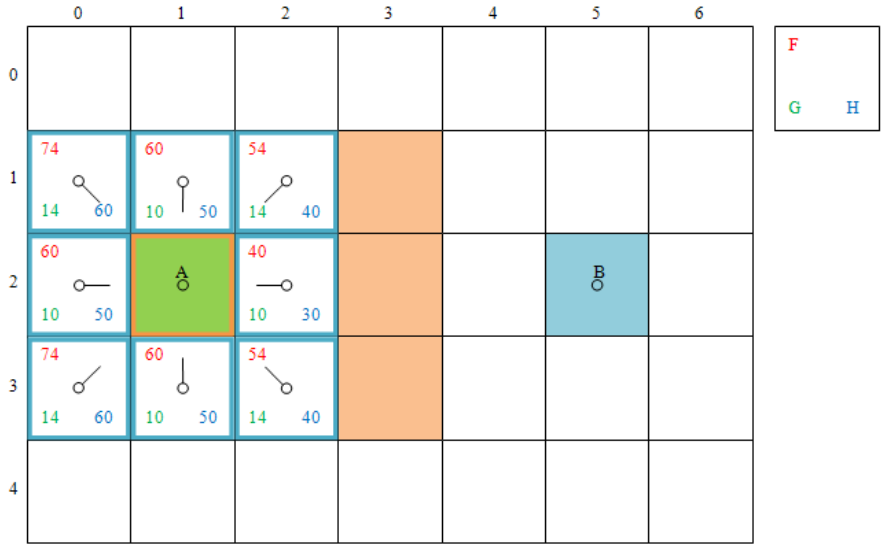


Рис. 4. Зображення клітин з обчисленими метриками.

Клітини, розташовані ортогонально до стартової клітинки, мають значення G рівне 10, а клітини, розташовані діагонально до стартової клітинки - G рівне 14. Значення H дорівнює Манхеттенській віддалі від центру поточної клітини до центру цільової клітини B, помножене на 10. Наприклад, для клітини з індексами [2, 2], відстань від її центру до центру цільової - 3 клітини (рисунок 5).

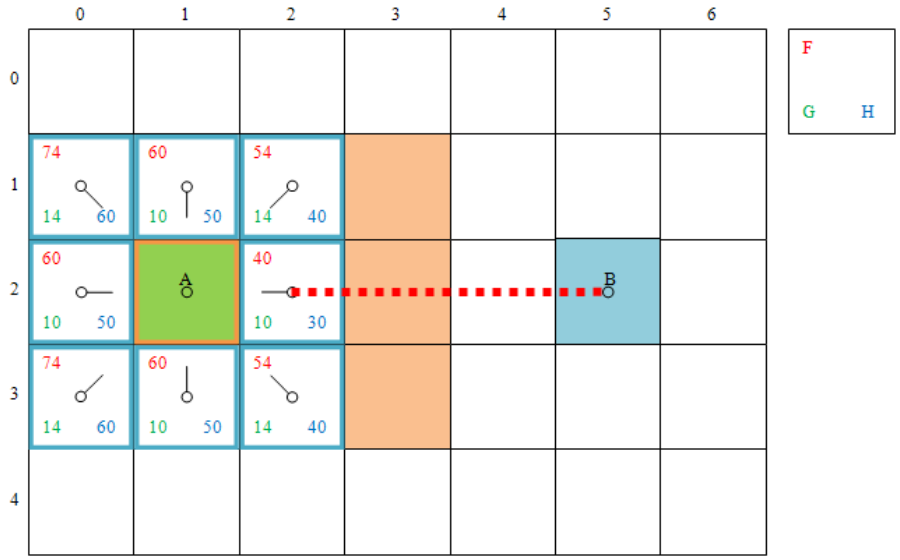


Рис.5. Відстань від центру клітини з індексами [2, 2] до центру цільової клітини.

А для клітин з індексами [1, 0] і [3, 0], відстань від їх центру до центру цільової клітини - 6 клітин (рисунок 6).

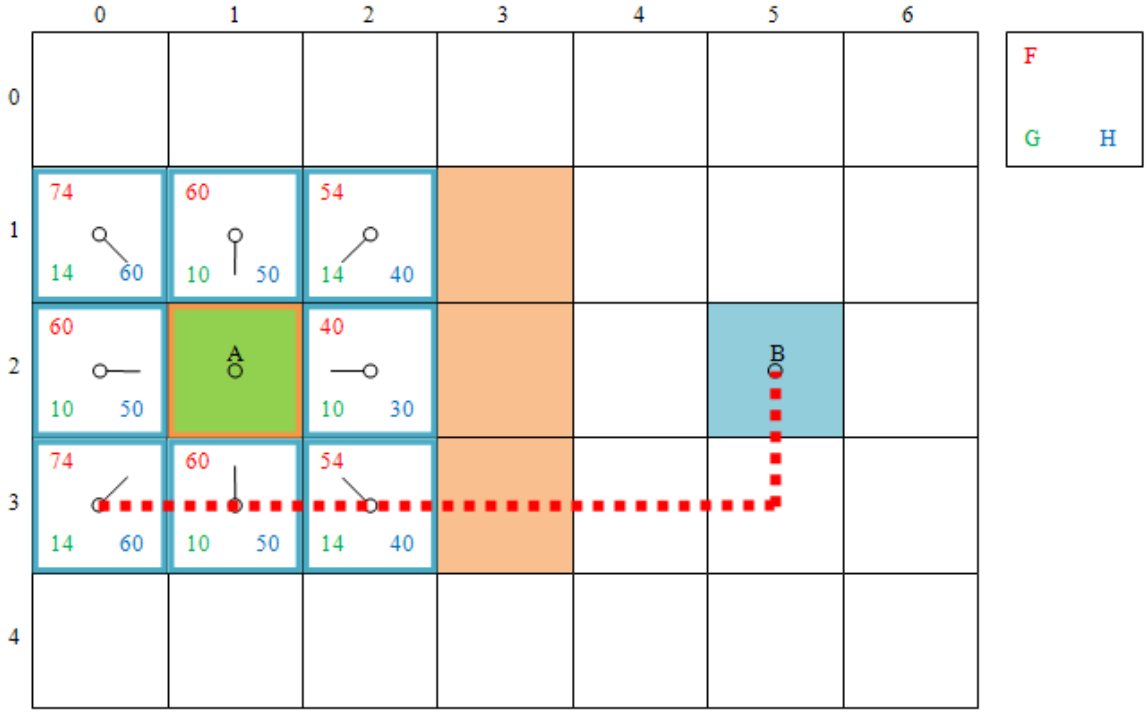


Рис.6. Відстань від центру клітини з індексами [1,0] і [3, 0] до центру цільової клітини.

Величина F для кожної клітини обчислюється за формулою 1, як сума величин G і H. Для продовження пошуку найкоротшого шляху вибирається комірка, з відкритого списку, з найменшим значенням F. І для цієї комірки діє таким чином: (продовження дій описаних вище):

1. Обрану з відкритого списку клітку видаляємо з нього і додаємо в закритий список.
2. Додаємо у відкритий список всі сусідні з нею клітини, якщо вони ще не знаходяться в ньому (при цьому ігноруючи непрохідні клітини і клітини, які

містяться в закритому списку). Попередньо вказавши, що поточна клітина є батьківською для клітин, доданих у відкритий список, а також обчисливши їх значення G, H і F.

3. Якщо сусідня клітина вже знаходиться у відкритому списку, то порівнюємо значення величин G у клітини з відкритого списку і клітини що перевіряється в даний момент. Якщо попереднє значення (у відкритому списку) менше нового, то нічого не робимо. У протилежному випадку, у клітини з відкритого списку міняємо значення G на нове, також міняємо вказівник на батьківську клітину так, щоб він вказував на поточну клітку, що перевіряється в даний момент.

Розглянемо описані кроки. Зараз у відкритому списку знаходиться 8 клітин, а стартова клітина - в закритому списку. З відкритого списку наступної кліткою для розгляду буде обрана клітка з найменшим значенням F - це комірка з індексами [2, 2], рисунок 7.

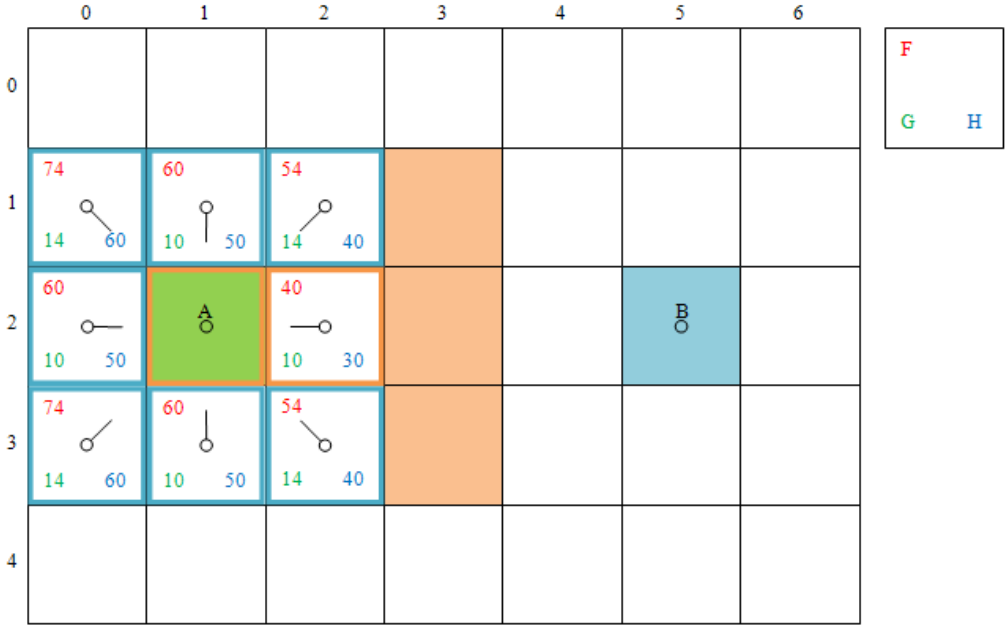


Рис.7. Вибір наступної клітини з відкритого списку для розгляду.

Спочатку поточну клітку (з індексами [2, 2]) видаляємо з відкритого списку, і поміщаємо в закритий список (тому на рисунку 7 вона відзначена оранжевим кольором). Потім перевіряємо сусідні клітини. Чотири з них, ігноруємо, - це три непрохідні клітини стіни і стартова клітина, яка перебуває в закритому списку. Чотири клітини вже розташовані в відкритому списку, а значить необхідно порівняти їх значення G зі значенням G таким, що якби ми до них дійшли через поточну клітку. Значення G у клітини нижче поточної (з індексами [3, 2]) дорівнює 14, а значення G, отримане при проході через поточну клітку дорівнює 20 (G = 10 у поточній комірці, і плюс 10 шлях до комірки нижче поточної). Значення 14 менше 20, тому значення G у комірки нижче поточної не потрібно оновлювати. У комірки зліва внизу (з індексами [3, 1]) G = 10, а значення G, отримане при проході через поточну клітку 24 (G = 10 у поточній комірці, і плюс 14 - шлях від поточної до комірки що перевіряється). 24 більше 10, значить оновлювати значення G у цій клітини не потрібно.

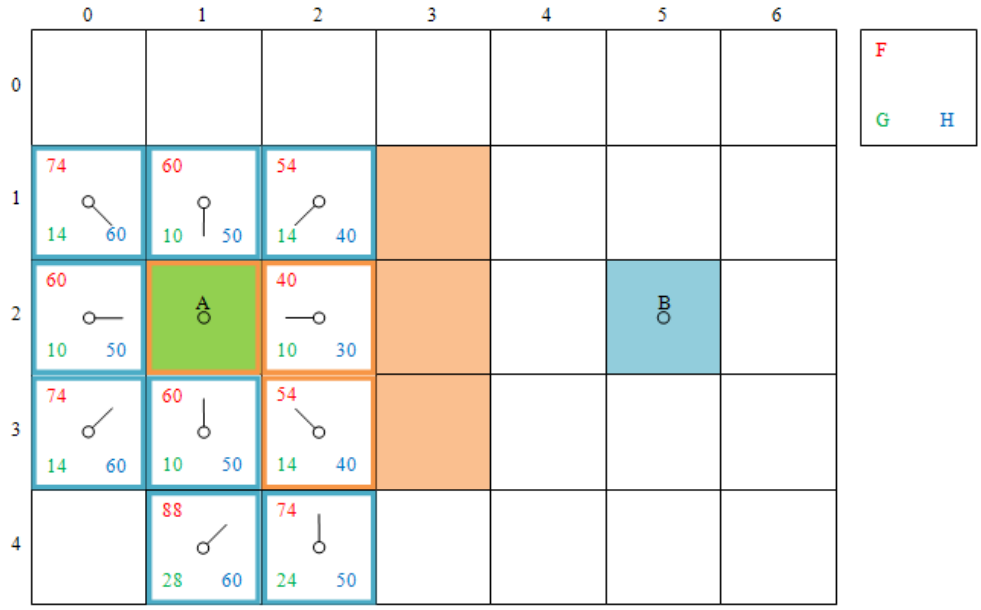


Рис.8. Вибір наступної клітини з індексами [3, 2].

Відповідно робляться перевірки і для клітини вище поточної (з індексами [1, 2]), і для клітини зліва вгорі від поточної клітини (з індексами [1, 1]). Після того, як всі сусідні клітини розглянуті, можна рухатися далі. На даний момент у відкритому списку знаходяться 7 клітин, дві з яких мають однакове найменше значення F рівне 54. Яку комірку вибрати наступною поточною з цих двох клітин, для алгоритму не має значення, тому уявимо, що випадковим чином вибрали комірку яка знаходиться справа внизу від стартової клітини (з індексами [3, 2]), як показано на рисунку 8.

Перевіряючи клітини з індексами [2, 3] і [3, 3], сусідні до поточної комірки, ігноруємо, оскільки вони непрохідні. Комірку з індексами [2, 2] і стартову клітку також ігноруємо - вони знаходяться в закритому списку. Клітку з індексами [4, 3] також ігноруємо, через те, що до неї неможливо дістатися без зрізу кута найближчої стіни. Спочатку необхідно перейти на клітку з індексами [4, 2], а потім вже переходити на клітку з індексами [4, 3]. (Правило забороняє зрізати кути у перешкод - необов'язкове до виконання, його застосування залежить від розташування Ваших вершин.) Клітка з індексами [3, 1] вже знаходиться у відкритому списку, тому порівнюємо її значення F зі значенням F таким, що якби ми прийшли на неї через поточну клітку. Це значення 60 і 64, відповідно, а значить, дані клітини, що перевіряються, не потрібно оновлювати. Клітини з індексами [4, 1] і [4, 2] додаємо у відкритий список, попередньо обчисливши їх значення величин G, H і F, а також встановивши покажчик на батьківську клітку (як проілюстровано на рисунку 8). Повторюємо описану вище методику пошуку шляху до тих пір, поки не додамо цільову клітку у відкритий список. Наступна клітина у відкритому списку з найменшим значенням F, рівним 54, це клітка з індексами [1, 2]. Видаляємо її з відкритого списку, додаємо в закритий список, та перевіряємо її сусідів (при цьому додаємо дві клітини з індексами [0, 1] і [0, 2]). Підсумок операцій проілюстровано на рисунку 9.

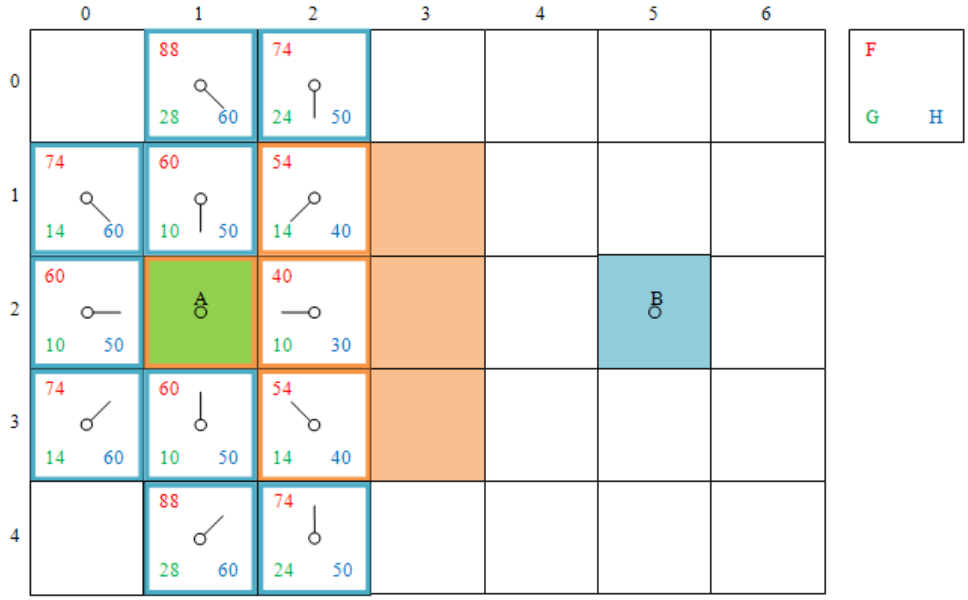


Рис.9. Підсумок проведених операцій.

Зараз у відкритому списку знаходиться дві клітини з найменшим значенням F рівним 60. Випадково вибираємо клітину з індексами [3, 1]. Видаляємо її з відкритого списку, додаємо в закритий список. Перевіряємо її сусідів (додаємо у відкритий список клітку з індексами [4, 0] і у клітини з індексами [4, 1], що вже перебуває у відкритому списку, оновлюємо значення F і міняємо вказівник на батька, тепер він посилається на поточну клітку, з індексами [3, 1]), рисунок 10.

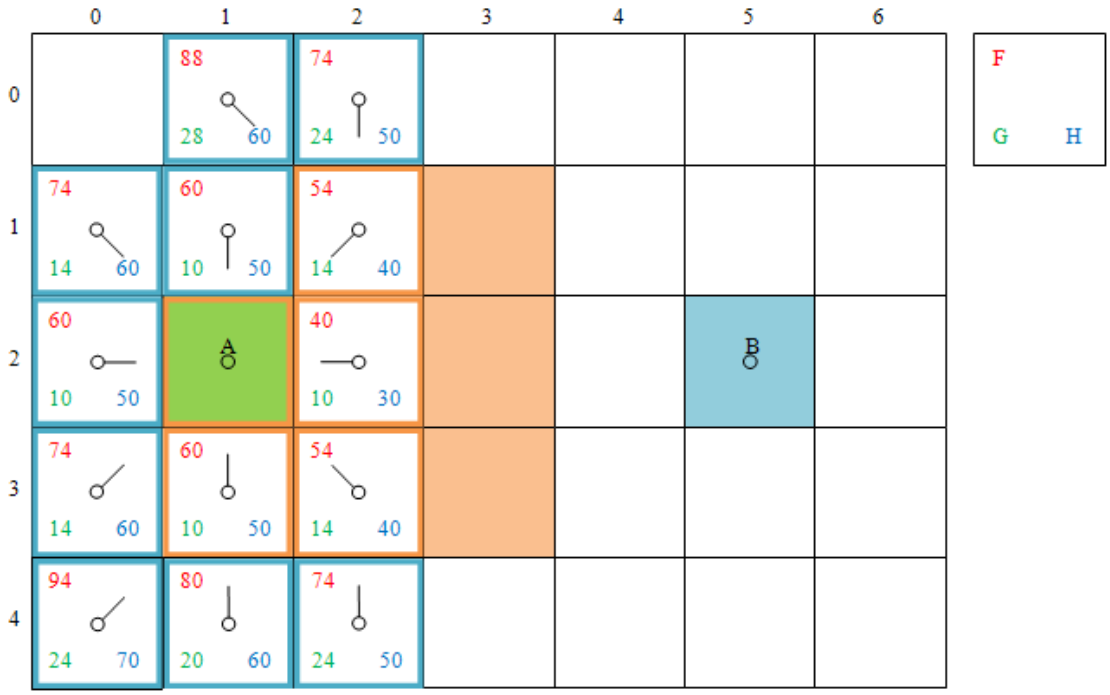


Рис.10. Вибір клітини з індексами [3, 1].

Наступну з відкритого списку опрацьовуємо клітку з індексами [2, 0]. Видаляємо її з відкритого списку і додаємо в закритий список. При цьому ніякі з її сусідніх клітин не потребують оновлення, рисунок 11.

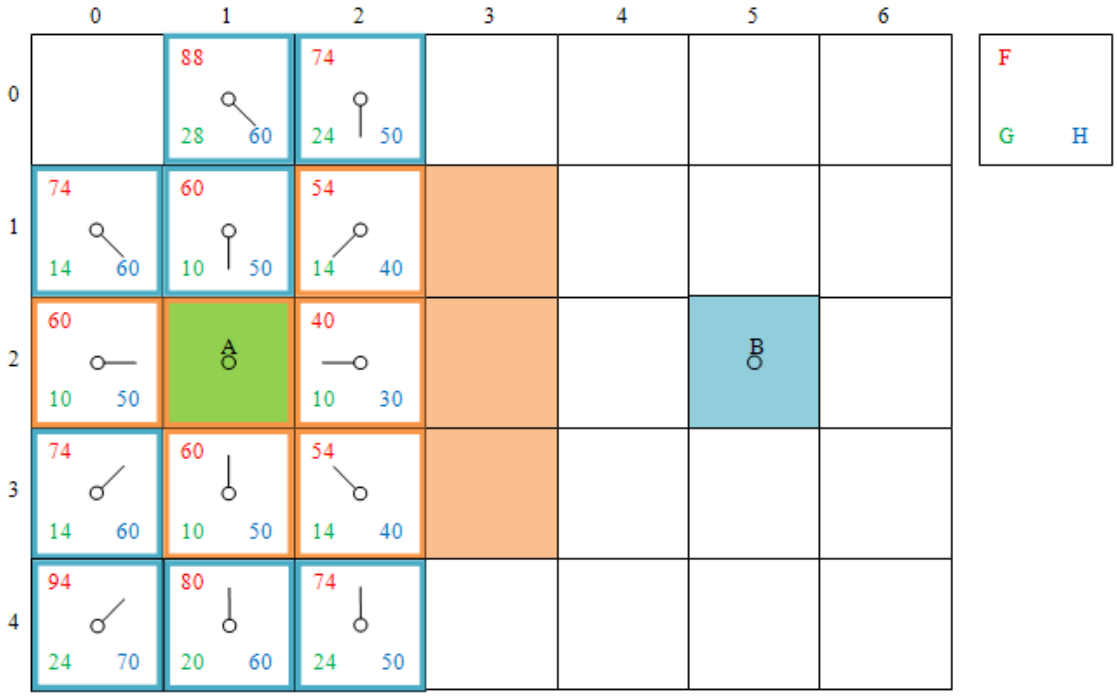


Рис.11. Вибір клітини з індексами [2, 0].

Далі опрацьовуємо клітку з індексами [1, 1]. Видаляємо її з відкритого списку, додаємо в закритий список. При цьому оновлюємо дані її сусідньої клітки з індексами [0, 1] і додаємо у відкритий список клітку з індексами [0, 0], рисунок 12.

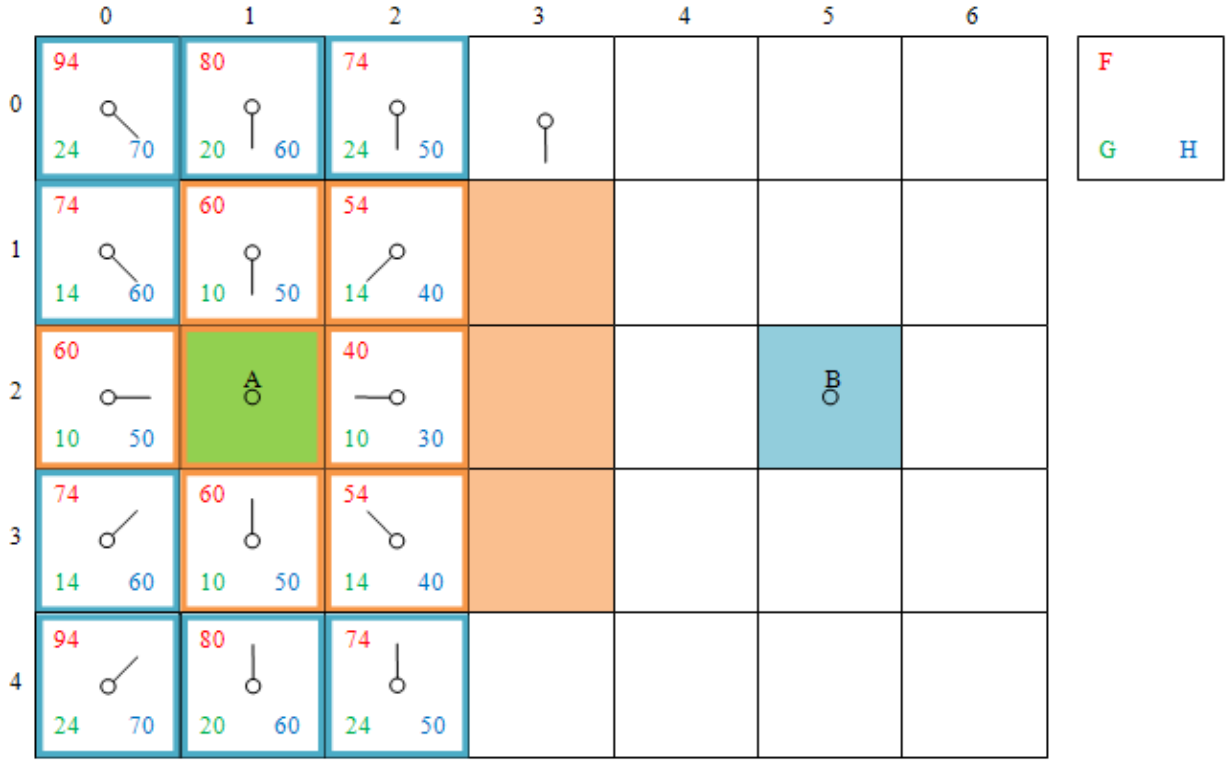


Рис.12. Вибір клітини з індексами [1, 1].

Наступні клітини у відкритому списку мають найменше значення F рівне 74. Випадковим чином вибираємо клітину з індексами [4, 2]. Видаляємо її з

відкритого списку, додаємо в закритий список. Перевіряємо її сусідів - додаємо у відкритий список клітку з індексами [4, 3], рисунок 13.

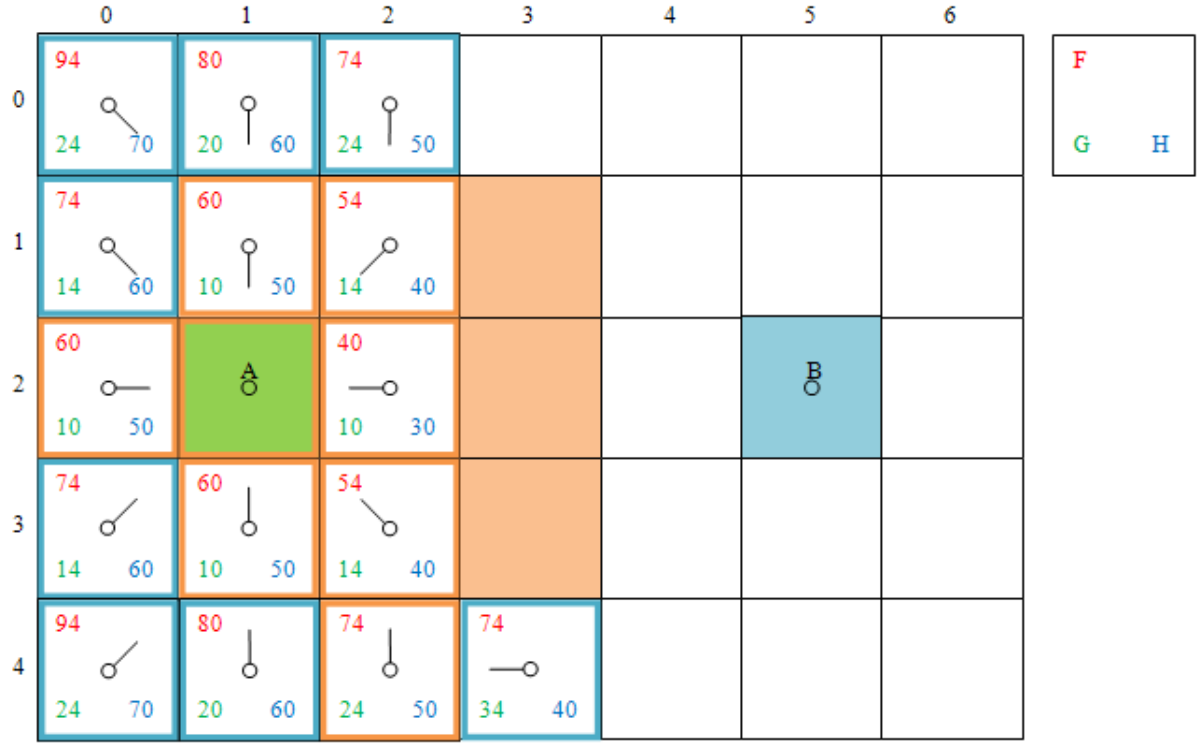


Рис.13. Вибір клітини з індексами [4, 2].

Припустимо, що наступною кліткою з найменшим значенням F обрана клітка ближче до тієї, яку перевіряли перед цим - клітина з індексами [4, 3]. Видаляємо її з відкритого списку, додаємо в закритий список. Опрацьовуємо її сусідні клітини (рисунок 14).

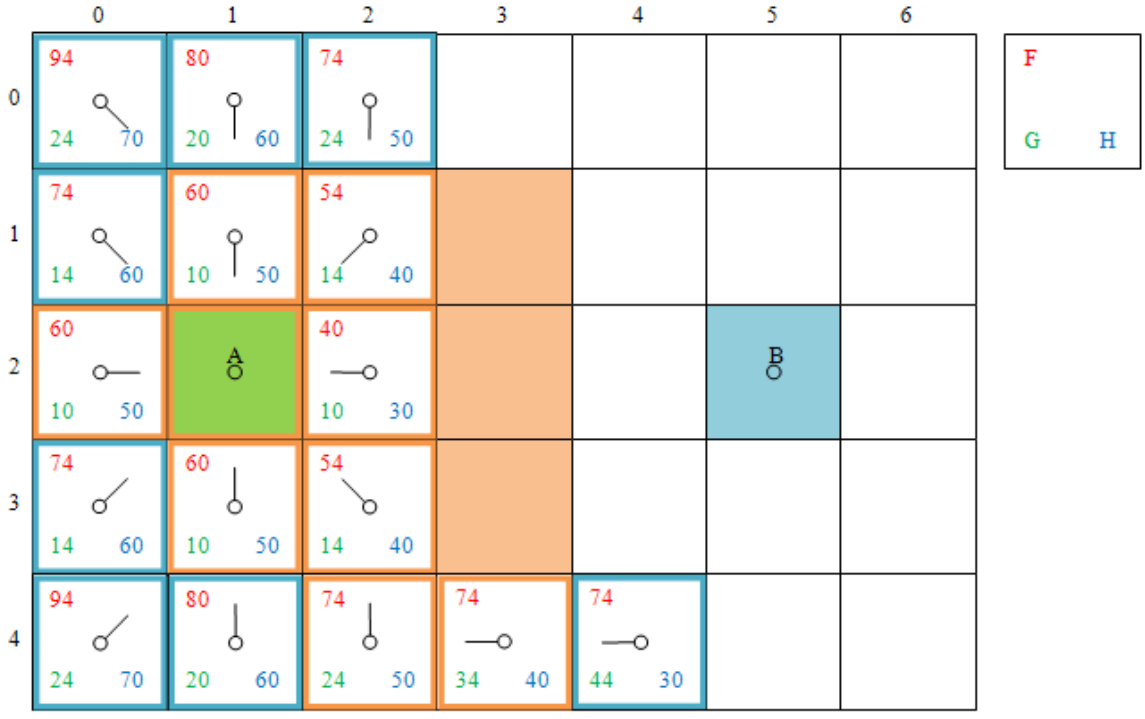


Рис.14. Вибір клітини з індексами [4, 3].

Далі, вибираємо клітину з індексами [4, 4], видаляємо її з відкритого списку, додаємо в закритий список. Опрацьовуємо її сусідні клітини - додаємо три клітини у відкритий список (клітини з індексами [3, 4], [3, 5] і [5, 4]) (рисунок 15).

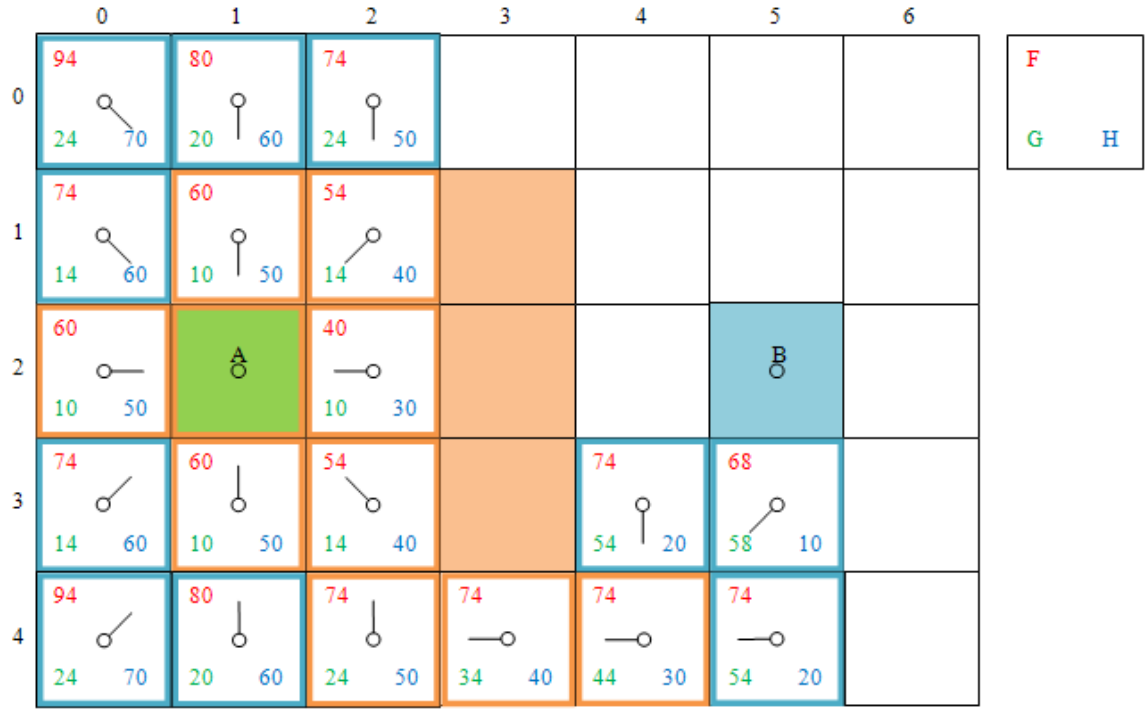


Рис.15. Вибір клітини з індексами [4, 4].

Наступна клітина з найменшим значенням F це клітина з індексами [3, 5]. Вибираємо її, видаляємо з відкритого списку, додаємо в закритий список. Опрацьовуємо її сусідні клітини, при цьому у відкритий список додаються клітини з індексами [2, 4], [3, 6], [2, 6] і цільова клітина з індексами [2, 5], рисунок 16.

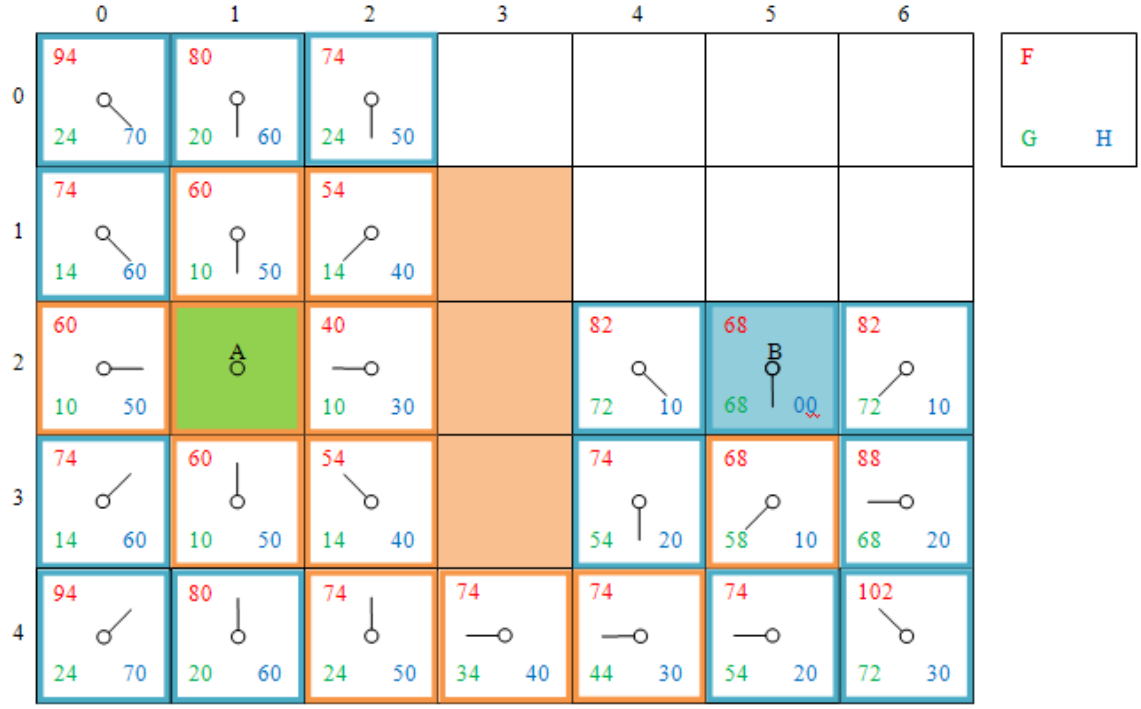


Рис.16. Вибір клітини з індексами [3, 5].

Цільова клітина знаходиться у відкритому списку, а це означає, що був знайдений шлях від стартової до фінішної клітини. Тепер слідуючи вказівниками на батьків можна пройти від фінішної клітини до стартової клітини, а збережений шлях у зворотному напрямку - шлях від стартової до цільової клітини, це і буде знайдений найкоротший шлях (рисунок 17).

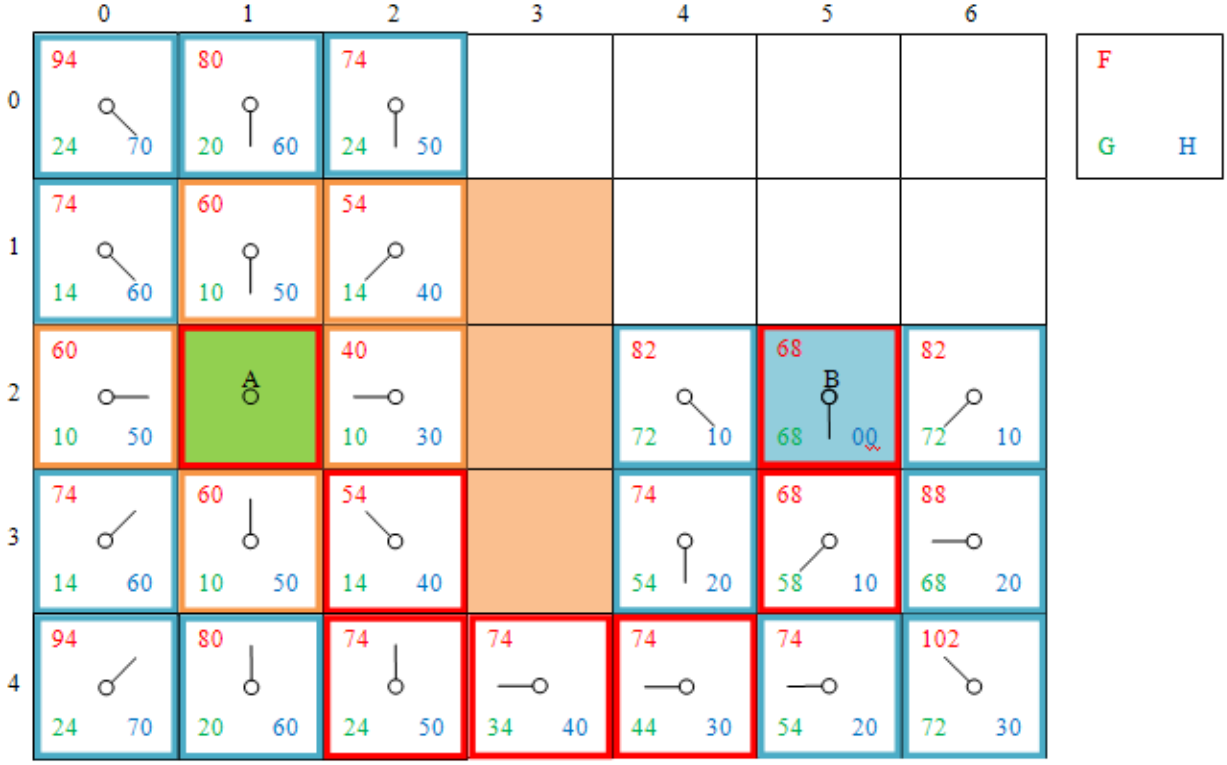


Рис.17. Вибір цільової клітини.

Словесний опис алгоритму:

1. Додати стартову клітку у відкритий список (при цьому її значення G, H і F рівні 0).
2. Повторювати наступні кроки:
   2.1. Шукаємо у відкритому списку клітку з найменшим значенням величини F, робимо її поточною.
   2.2. Видаляємо поточну клітину з відкритого списку і поміщаємо її в закритий список.
   2.3. Для кожної з сусідніх, до поточної клітини, кліток:
      2.3.1. Якщо клітина непрохідна або знаходиться в закритому списку, ігноруємо її.
      2.3.2. Якщо клітина не в відкритому списку, то додаємо її у відкритий список, при цьому розраховуємо для неї значення G, H і F, і також встановлюємо посилання батька на поточну клітку
      2.3.3. Якщо клітина знаходиться у відкритому списку, то порівнюємо її значення G зі значенням G таким, що якби до неї прийшли через поточну клітку. Якщо збережене в перевіреній клітці значення G більше нового, то міняємо її значення G на нове, перераховуємо її значення F і змінюємо вказівник на батька так, щоб вона вказувала на поточну клітку.
   2.4. Зупиняємося, якщо:
      2.4.1. У відкритий список додали цільову клітку (в цьому випадку шлях знайдений).
      2.4.2. Відкритий список порожній (в цьому випадку до цільової клітини шляху не існує).
3. Зберігаємо шлях, рухаючись назад від цільової клітини, проходячи за вказівниками на батьків до тих пір, поки не дійдемо до стартової клітини.

3. КОНТРОЛЬНІ ЗАПИТАННЯ

1. *Яку структуру даних найчастіше використовують для реалізації алгоритму A*?*
2. *Розказати алгоритм A*.*
3. *Намалювати блок-схему алгоритму A*.*
4. *Яка сітка використовується для алгоритму A*?*
5. *Яка основна формула використовується в алгоритмі A* для розрахунку пріоритету вузла?*
6. *Що означає параметр F(n) у контексті алгоритму A*?*
7. *Що означає параметр g(n) у контексті алгоритму A*?*
8. *Що означає параметр h(n) у контексті алгоритму A*?*
9. *Яка умова має виконуватися для евристичної функції h(n), щоб алгоритм A* був допустимим (гарантовано знаходив найкоротший шлях)?*
10. *Яку структуру даних найефективніше використовувати для списку відкритих вузлів (Open List) в A*?*
11. *У що перетворюється алгоритм A*, якщо евристична функція h(n) завжди дорівнює 0?*
12. *Яка евристика зазвичай використовується для сітки, де дозволено рух лише у чотирьох напрямках (вгору, вниз, вліво, вправо)?*
13. *Що відбувається, якщо евристика h(n) значно перевищує реальну вартість шляху до цілі?*
14. *Яка роль списку закритих вузлів (Closed List) в алгоритмі A*?*
15. *Якщо дві евристики h1(n) та h2(n) обидві є допустимими, і h2(n) > h1(n) для всіх вузлів, то яка з них краща?*
16. *Як A* відновлює фінальний шлях після досягнення цільового вузла?*

4. ЗАВДАННЯ ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Ще більше вдосконалити **(до рівня senior ;)** свої вміння грати в гру - “Гра у 8” з метою знаходження самого оптимальнішого рішення поставленого завдання. Записати у звіт 2-3 нові послідовності переміщень пустого квадрата відповідно до початкового стану Вашого варіанту для знаходження заданого кінцевого стану. Зафіксувати та записати у звіт час, потрачений Вами на знаходження кожного із цих нових рішень.

Дослідити рішення задачі “Гра в 8” за допомогою оціночної функції методом А*.

1. Розв'язати задачу “Гра в 8” за допомогою методу оціночної функції методом А* до третього рівня дерева пошуку (початковий стан – перший рівень) ручним способом. У звіті навести дерево пошуку.

2. Розробити алгоритм і програму рішення задачі “Гра в 8” за допомогою методу оціночної функції А*.

3. Узвіті навести кількість згенерованих станів, кількість станів занесених в базу станів, кількість відкинутих станів а також глибину дерева пошуку на якій знайдено рішення.

4. У звіті навести блок-схему алгоритму і роздрук програми.

5. Дослідження провести для наступних метрик оціночної функції.

- Евклідова відстань
- Відстань міських кварталів
- Відстань Чебишева

6. Для методу оціночної функції узагальнити проведені дослідження звівши на одній діаграмі для трьох метрик оціночної функції кількість згенерованих станів, кількість станів занесених в базу станів, кількість відкинутих станів.

7. Програмна реалізація повинна передбачати покрокове виконання програми і перехід на повне виконання програми.

8. Провести аналіз ефективності рішення задачі “Гра у 8” за допомогою алгоритму А* в порівнянні з неінформативними методами пошуку у ширину та глибину для свого варіанту завдання.

Потрібно покроково описати повний хід виконання Вашої лабораторної роботи: який пункт завдання виконуєте, чому саме таким методом, як його перевіряєте, отримали те що планувалось чи ні і т.д.

Вимоги до оформлення звіту

1. Титульний аркуш.
2. Тема.
3. Мета.
4. Теоретичні відомості.
5. Блок-схема алгоритму A*.
6. Словесний опис алгоритму A*.
7. Ручні обчислення.
8. Код програми з коментарями.
9. Автоматизовані обчислення.
10. Графічний результат виконання програми.
11. Результати аналізу ефективності рішення задачі “Гра у 8” за допомогою алгоритму A* в порівнянні з неінформативними методами пошуку у ширину та глибину для свого варіанту завдання
12. Аналіз результатів та помилок, допущених під час виконання лабораторної роботи.
13. Висновки.

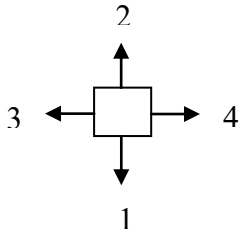
5. СПИСОК ЛІТЕРАТУРИ

1. Адіт’я Бхарґава. Грокаємо алгоритми. Ілюстрований посібник для програмістів і допитливих. ArtHuss. 2023. 256 с.
2. Олексій Васильєв. Алгоритми. Ліра-К. 2022. 424 с.
3. Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Стайн. Вступ до алгоритмів. MIT Press. 2019. 1289 с.
4. Robert Sedgewick, Kevin Wayne. Algorithms (4th Edition). Addison-Wesley. 2011. 956 p.
5. Heinemann George. Algorithms. With Examples in Python. O'Reilly. 2024. 304 p.

6. ВАРІАНТИ ЗАВДАНЬ

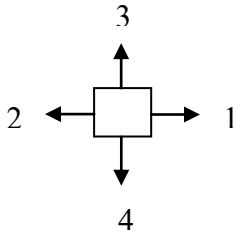
Варіант №1.

□	5	6
7	8	1
2	3	4



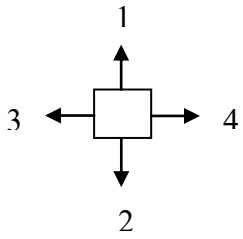
Варіант №2.

5	6	
7	8	1
2	3	4



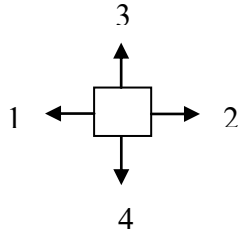
Варіант №3.

5	6	7
8		1
2	3	4



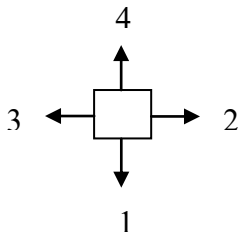
Варіант №4.

5	6	7
8	1	2
	3	4



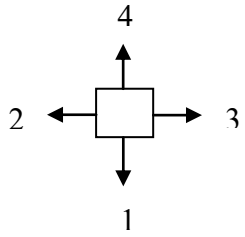
Варіант №5.

5	6	7
8	1	2
3	4	



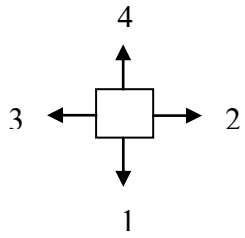
Варіант №6.

□	7	8
4	5	6
1	2	3



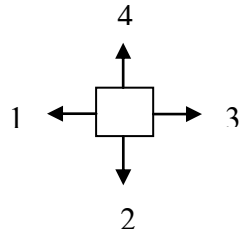
Варіант №7.

7	8	
4	5	6
1	2	3



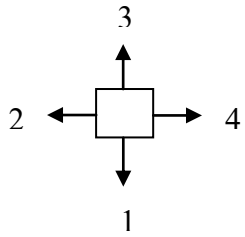
Варіант №8.

7	8	4
5		6
1	2	3



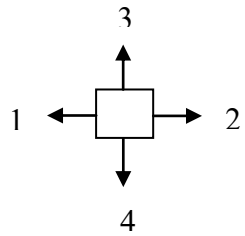
Варіант №9.

7	8	4
5	6	1
	2	3



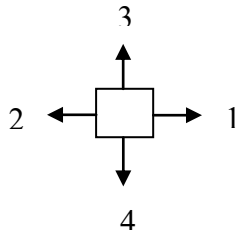
Варіант №10.

7	8	4
5	6	1
2	3	



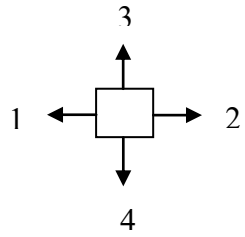
Варіант №11.

□	3	4
1	2	6
7	5	8



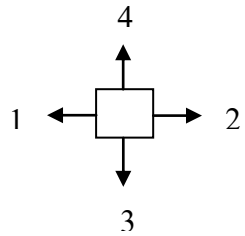
Варіант №12.

3	4	
1	2	6
7	5	8



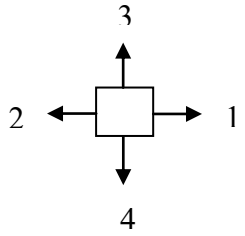
Варіант №13.

3	4	1
2		6
7	5	8



Варіант №14.

3	4	1
2	6	7
	5	8



Варіант №15.

3	4	1
2	6	7
5	8