1 ОПИС АЛГОРИТМІВ

Перелік всіх основних змінних та їхнє призначення наведено в таблиці 3.1.

Таблиця 3.1 – Основні змінні та їхні призначенням

Змінна	Призначення
seq	послідовність дій для розв’язку задачі
reord_cell	Зберігає клітинку, яку користувач може поміняти місцями при створенні початкового поля вручну
along	Напрям уздовж
across	Напрям перпендикулярно
top	Позиція, що відповідає передостанньому рядку на третьому з кінця стовпцю
bot	Позиція, що відповідає останньому рядку на третьому з кінця стовпцю
p	Масив попередніх вершин у шляху
v	Поточна вершина

1.1. Загальний алгоритм

1. ПОЧАТОК
2. Зчитати спосіб задавання початкового стану поля.
   - 2.1. ЯКЩО генерація випадковим чином ТО викликати метод випадкової генерації
   - 2.2. ЯКЩО створення вручну ТО викликати метод зміни порядку користувачем
3. Створити масив seq
4. ДОПОКИ seq не порожній
   - 4.1. ЯКЩО користувач наниснув на клітинку поруч із порожньою ТО перемістити її на порожнє місце

4.1.1. видалити перший елемент seq

4.1.2. ЯКЩО переміщення не відповідає йому ТО створити новий масив seq

4.2. ЯКЩО користувач натиснув на кнопку наступного кроку ТО виконати наступне переміщення розв’язку

4.3. ЯКЩО користувач натиснув на кнопку автоматичного розв’язування

4.3.1. ДОПОКИ seq не порожній ТА користувач не натискає кнопку зупинки виконувати наступне переміщення розв’язку

5. КІНЕЦЬ

1.2. Алгоритм випадкової генерації поля

1. ПОЧАТОК
2. Створити випадково згенерований масив
3. ЯКЩО парність рядка пробіла та парність кількості інверсій решти клітинок збігаються ТО змінити місцями пробіл та клітинку поруч із сусідньою у тому ж рядку
4. КІНЕЦЬ

1.3. Алгоритм зміни розташування поля користувачем

1. ПОЧАТОК
2. ДОПОКИ користувач не натисне кнопку почати гру ТА поле буде відмінне від виграшного ТА його можна буде розв’язати

2.1. ЯКЩО користувач натиснув на клітинку

2.1.1. ЯКЩО reord_cell зберігає клітинку ТО поміняти її місцями з попередньою та очистити reord_cell

2.1.2. ІНАКШЕ зберегти цю клітинку в reord_cell

3. КІНЕЦЬ

1.4. Алгоритм створення масиву послідовності дій для розв’язання пазла

1. ПОЧАТОК

2. Створити граф, в якому вершини відповідають позиціям у полі, а ребра поєднують ті із них, що мають спільні сторони
3. Послідовно заповнити відповідними значеннями верхній рядок, лівий стовпець та другий рядок згори, додаючи переміщення пробілу у seq
4. Заповнити відповідними значеннями останні п’ять позицій, додаючи усі переміщення пробілу у seq
5. КІНЕЦЬ

1.5. Алгоритм заповнення рядка або стовпця відповідними значеннями

1. ПОЧАТОК
2. ЯКЩО заповнюється рядоок ТО присвоїти along напрям вправо, присвоїти across напрям вниз
3. ЯКЩО заповнюється стовпець ТО присвоїти along напрям вниз, присвоїти along напрям вправо
4. ЯКЩО лінія заповнена ТО видалити із графу вершини, що відповідають позиціям лінії та завершити виконання функції
5. ІНАКШЕ
    5.1. ЦИКЛ перебору всіх клітинок рядка, починаючи із напрямка протилежного along, окрім останньої
        5.1.1. Перемістити клітинку на своє місце
        5.1.2. Видалити відповідну вершину із графу
    5.2. Перемістити останню клітинку у відповідну їй позицію, зміщену у напрямку across
    5.3. Перемістити пробіл на позицію, зміщену на across від другої з кінця у лінії, що заповнюється, не переміщуючи при цьому останню клітинку
    5.4. Виконати послідовність переміщень для пробіла: -across, along, across, -along, -across, -along, across
    5.5. Видалити із графу вершину, що відповідає останній клітинці у лінії
6. КІНЕЦЬ

1.6. Алгоритм заповнення двох лівих позицій із прямокутника 2 на 3, що залишився після виконання функцій, що заповнюють рядки зверху та стовпці зліва

1. ПОЧАТОК
2. ЯКЩО ці дві клітинки на на своїх місцях
    2.1. Присвоїти змінній bot значення позиції нижньої із них
    2.2. Присвоїти змінній top значення позиції верхньої із них
    2.3. Перемістити клітинку, що відповідає bot у позицію top
    2.4. Перемістити пробіл на позицію справа від позиції bot
    2.5. ЯКЩО у позиції bot знаходиться клітика, що відповідає top ТО виконати наступну послідовність переміщень для пробіла: вліво, вгору, вправо, вниз, вправо, вгору, вліво, вліво, вниз
    2.6. Перемістити клітинку, що відповідає top у позицію справа від top
    2.7. Перемістити клітинку, що відповідає позиції bot, у відповідну їй позицію, не переміщуючи клітинку, що відповідає top
    2.8. Перемістити пробіл вправо, поставивиши тим самим у top відповідне значення
3. Видалити з графу вершини, що відповідають top та bot
4. КІНЕЦЬ

1.7. Алгоритм заповнення останніх трьох клітинок

1. ПОЧАТОК
2. Заповнити позицію зліва та зверху від правого нижнього кута поля відповідною клітинкою
3. Заповнити позицію зверху від правого нижнього кута поля відповідною клітинкою
4. Заповнити позицію зліва від правого нижнього кута поля відповідною клітинкою
5. КІНЕЦЬ

1.8. Алгоритм переміщення клітинки у певну позицію

1. ПОЧАТОК
2. Знайти найоротший шлях від поточної клітинки до кінцевої позиції, використовуючи пошук вшир у графі
3. ЦИКЛ перебору усіх переходів у цьому шляху
    3.1. Виключити вершину, що відповідає поточній позиції клітинки, що переміщується, із графу
    3.2. Перемістити пробіл у поточну позицію у шляху
    3.3. Повернути останню вилучену вершину у граф
    3.4. Перемістити пробіл на місце клітинки, що переміщується
4. КІНЕЦЬ

1.9. Алгоритм переміщення пробіла у певну позицію

1. ПОЧАТОК
2. Знайти найоротший шлях від пробіла до кінцевої позиції, використовуючи пошук вшир у графі
3. ЦИКЛ перебору усіх переходів у цьому шляху
    3.1. перемістити пробіл у поточну позицію шляху
4. КІНЕЦЬ

1.10. Алгоритм знаходження найкоротшого шляху між вершинами у графі за доппомогою пошуку вшир

1. ПОЧАТОК
2. Створити чергу
3. Створити масив p попередніх вершин у шляху для кожної вершини та позначити їх як непройдені
4. Відмітити початкову вершину, як таку, що не має попередньої
5. Ініціалізувати поточну вершину v початковою вершиною
6. ДОПОКИ v не є кінцевою вершиною
    6.1. ЦИКЛ перебору сусідніх вершин зі списку суміжності для v
        6.1.1. ЯКЩО сусідня вершина не перевірена ТО додати її у кінець черги та вказати вершину v як попередню їй

6.2. Замінити v на вершину, вилучену з початку черги

7. Створити масив результату
8. ДОПОКИ v не є початковою вершиною
   8.1. Додати в кінець масиву результату v
   8.2. Замінити вершину v на попередню для неї
9. Змінити порядок у масиві результату на зворотній
10. КІНЕЦЬ