

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ
З ПРАКТИЧНОЇ РОБОТИ №4
дисципліна: «Алгоритми та структур и даних»

Виконала: студентка групи КС-22

Узенкова Дар'я

Перевірив: Олешко Олег

Харків

2024

Завдання. Напишіть програму для рішення головоломки.

Головоломка представляє собою масив цілих чисел, наприклад:

3 6 4 1 3 4 2 5 3 0

Зелений колір в першій клітинці вказує ваше теперішнє положення. На кожному етапі головоломки ви можете рухатися рівно на стільки клітинок, скільки вказано цілим числом в клітинці, в якій ви знаходитесь. Ви можете рухатися або вліво, або вправо, але не можете виходити за масив. Наприклад, єдиним дозволеним першим кроком є переміщення на три клітинки вправо, тому що немає місця для переміщення на три клітинки вліво. Мета головоломки - перемістити зелений маркер на 0 в останній клітинці. В наведеній конфігурації ви можете вирішити головоломку, зробивши такі кроки:

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

Незважаючи на те, що для наведеного прикладу ця головоломка може бути розв'язана — і, дійсно, має більше одного рішення, — деякі головоломки цього виду можуть не мати рішення взагалі, наприклад, наступна:

3 1 2 3 0

В цій головоломці ви будете рухатися між двома трійками, але не зможете потрапити до інших клітинок.

Пояснення до завдання

1. Всі числа в масиві є позитивними.
2. Змінювати значення в масиві заборонено.
3. Останній елемент масиву дорівнює 0. (Взагалі кажучи, якщо ви туди дісталися, не важливо, яке число там зберігається.)
4. Інших нулів в масиві немає. (Взагалі кажучи, якщо будуть інші нулі и ви потрапите в таку клітинку, ви в ній і застрягнете.)
5. Існує дозволений перехід із кожної клітинки масиву. (Уявіть, якщо в одній із клітинок вищевказаного прикладу буде число 20.)

Вимоги до програми

Програма заповнює масив розміром 10 випадковими значеннями (не більше 9), виводить на екран вихідний масив, а потім всі кроки рішення задачі та ВСІ варіанти рішення в вигляді:

3 R 1 R 3 L 6 R 5 L 4 R 2 R 3 L 4 R 0 (де вказується число в клітинці та напрям руху «R/L» — right/left).

Результати виконання завдання 1 наведено:

1. У лістингу 1 — вихідний код програми.
2. У малюнках 1, 2, 3, 4 — результати виконання програми.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define SIZE 10

// Функція для пошуку з поверненням
void FindAPath(int arr[], int position, char *path, int *found, int
visited[]) {
    // Якщо досягли кінцевої позиції
    if (position == SIZE - 1) {
        printf("%s0\n", path); // Додаємо 0 до шляху
        *found = 1; // Відзначаємо, що знайдено хоча б одне рішення
        return;
    }

    int step = arr[position];

    // Рух вправо
    if (position + step < SIZE && !visited[position + step]) {
        visited[position + step] = 1; // Помічаємо позицію як відвідану
        char newPath[100];
        snprintf(newPath, sizeof(newPath), "%s%d R ", path, step);
        FindAPath(arr, position + step, newPath, found, visited);
        visited[position + step] = 0; // Відмічаємо позицію як не відвідану
    }

    // Рух вліво
    if (position - step >= 0 && !visited[position - step]) {
        visited[position - step] = 1; // Помічаємо позицію як відвідану
        char newPath[100];
        snprintf(newPath, sizeof(newPath), "%s%d L ", path, step);
        FindAPath(arr, position - step, newPath, found, visited);
        visited[position - step] = 0; // Відмічаємо позицію як не відвідану
    }
}

int main() {
    srand(time(NULL));
    int arr[SIZE];

    // Генерація масиву випадкових чисел (не більше 9)
    for (int i = 0; i < SIZE - 1; i++) {
        arr[i] = rand() % 9 + 1;
    }
    arr[SIZE - 1] = 0; // Останній елемент завжди 0

    // Виведення масиву
    printf("Масив: ");
    for (int i = 0; i < SIZE; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n\nВаріанти рішень:\n");

    // Пошук всіх можливих рішень

```

```

char path[100] = ""; // Ініціалізація пустого шляху
int found = 0; // Прапор для перевірки, чи знайдено рішення
int visited[SIZE] = {0}; // Масив для відстеження відвіданих позицій
visited[0] = 1; // Починаємо з першої позиції

FindAPath(arr, 0, path, &found, visited);

// Якщо не знайдено жодного рішення
if (!found) {
    printf("Рішення не знайдено.\n");
}

return 0;
}

```

Лістинг 1 – вихідний код програми

```

Масив: 6 3 8 1 5 9 3 6 1 0

Варіанти рішень:
6 R 3 R 0
6 R 3 L 1 R 5 R 0

-----
Process exited after 0.03851 seconds with return value 0
Press any key to continue . . .

```

Малюнок 1 – результат виконання програми

```

Масив: 5 5 7 5 1 3 4 8 5 0

Варіанти рішень:
5 R 3 L 7 R 0

-----
Process exited after 0.044 seconds with return value 0
Press any key to continue . . .

```

Малюнок 2 – результат виконання програми

```

Масив: 2 5 2 6 6 7 1 7 6 0

Варіанти рішень:
Рішення не знайдено.

-----
Process exited after 0.04321 seconds with return value 0
Press any key to continue . . .

```

Малюнок 3 – результат виконання програми