

РЕШЕНИЕ ГОЛОВОЛОМКИ НА ПОЛЕ СЛОВ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА НАХОЖДЕНИЯ МИНИМАЛЬНОГО ПУТИ

Введение. Филворды — это популярная головоломка, в которой игроку предоставляется квадратное поле с буквами, и задачей является нахождение всех заданных слов на этом поле. Данная статья представляет алгоритм создания этой головоломки с использованием метода нахождения минимального пути [1]. Программа, реализующая данный алгоритм, создает поле слов, заполняет его буквами на заданном поле.

Основная часть. Программа использует следующие структуры данных:

1. `WordField`: Класс, представляющий поле слов и содержащий методы для его инициализации, отображения и решения.

2. `point`: Структура, представляющая координаты на поле.

3. `pointHash`: Функция хэширования для хранения путей в хэш-таблице.

4. `globalPaths`: Хэш-таблица для хранения найденных путей.

5. `freeCells`: Вектор, содержащий свободные ячейки на поле.

При создании экземпляра класса `WordField` происходит инициализация поля размером $N \times N$, где N задается пользователем. Заполняются два двумерных массива `LETFLD` и `VALFLD`. `LETFLD` представляет собой поле букв, и каждая ячейка может содержать либо букву, либо символ '*', обозначающий свободную ячейку. `VALFLD` содержит числовые значения для каждой ячейки, которые используются для оценки веса пути.

Для нахождения минимального пути для каждого заданного слова программа использует рекурсивный алгоритм `findPath`. Начиная с определенной точки на поле, алгоритм исследует все возможные направления (вверх, вниз, влево и вправо), чтобы найти путь, который соответствует заданной длине слова. Путь сохраняется в виде вектора структур `point`, который представляет координаты ячеек на поле.

Для каждого найденного пути программа вычисляет вес пути с помощью функции `countWeigh`. Вес пути рассчитывается как сумма значений в соответствующих ячейках `VALFLD`.

После нахождения всех возможных путей для заданного слова, программа выбирает путь с минимальным весом с помощью функции `getMinPath`.

После нахождения минимального пути для слова, программа заполняет соответствующие ячейки на поле буквами из заданного слова с использованием функции `refillLetfld`. После заполнения слова на поле, значения `VALFLD` обновляются с учетом нового состояния поля с использованием функции `refillValfld`.

Класс `WordField` имеет метод `Set`, который выполняет следующие шаги:

1. Выбор слова для размещения на поле.
2. Поиск свободных ячеек на поле.
3. Поиск минимального пути для выбранного слова.
4. Заполнение слова на поле.
5. Обновление значений `VALFLD`.
6. Очистка временных структур данных.
7. Повторение процесса для следующего слова.

Алгоритм может использоваться для создания головоломок на полях слов, таких как филворды или головоломки "найди слово". Он автоматически находит и заполняет слова на поле, минимизируя общий вес пути. Это позволяет создавать головоломки быстро и эффективно.

Однако алгоритм имеет некоторые ограничения и проблемы:

1. Производительность зависит от размера поля и количества слов. На больших полях или с большим количеством слов алгоритм может работать долго.

2. Возможность нахождения оптимального решения зависит от случайности инициализации значений в `VALFLD`. В некоторых случаях может существовать несколько оптимальных путей, и алгоритм выбирает один из них.

3. Алгоритм заполняет поле не полностью, оставляя пустые области, которые можно инициализировать случайными буквами.

Заключение. Алгоритм предоставляет создание головоломки на поле слов с использованием метода нахождения минимального пути. Он эффективно находит и заполняет слова на поле, оптимизируя вес пути. Несмотря на некоторые ограничения и случайные аспекты, этот алгоритм может быть полезным инструментом для создания подобных головоломок.

Список цитируемых источников

1. *Перельман, Я. И.* Головоломки и развлечения / Я. И. Перельман. — Аванта, 2020 г. — 224 с.