

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЯ» Институт заочного образования

Выполнила: Белова Ксения Юрьевна

Описание кода:

Данный код на Python реализует алгоритм поиска кратчайшего пути между двумя словами, отличающимися только одной буквой, используя представление слов в виде графа и алгоритм поиска в ширину (BFS).

Функциональные компоненты:

- **`create_word_graph(words)`**: Эта функция создает граф, где узлы представляют собой слова из заданного списка `words`. Ребра в графе соединяют два слова, если они отличаются друг от друга только одной буквой. Функция возвращает словарь, представляющий граф. Ключами словаря являются слова, а значениями – списки соседних слов, отличающихся от данного слова только одной буквой.
- **`edit_distance_one(word1, word2)`**: Эта функция определяет, отличаются ли два слова `word1` и `word2` только одной буквой. Если длины слов не совпадают, функция возвращает `False`. В противном случае функция сравнивает буквы в словах попарно и подсчитывает количество различий. Если количество различий равно 1, функция возвращает `True`, иначе – `False`.
- **`shortest_path(graph, start, end)`**: Эта функция находит кратчайший путь между двумя заданными словами (`start` и `end`) в заданном графе `graph`, используя алгоритм поиска в ширину (BFS). Алгоритм BFS гарантирует нахождение кратчайшего пути в графе, если такой путь существует. Функция возвращает список слов, представляющий найденный кратчайший путь от `start` до `end`. Если путь не найден, функция возвращает `None`.

Алгоритм работы:

1. **Создание графа**: Сначала создается граф слов с помощью функции `create_word_graph(words)`. Функция `create_word_graph` создает граф на основе списка предоставленных слов, где соседними словами являются те, которые отличаются ровно одной буквой.
2. **Поиск кратчайшего пути**: Затем используется функция `shortest_path(graph, start_word, end_word)` для поиска кратчайшего пути между двумя заданными словами (`start_word` и `end_word`) в созданном графе. Функция `shortest_path` использует алгоритм BFS для нахождения кратчайшего пути.

3. **Вывод результата:** Если кратчайший путь найден, программа выводит найденный путь на консоль. В противном случае программа выводит сообщение о том, что путь между заданными словами не найден.

Используемые структуры данных и алгоритмы:

- **Граф:** Словарь, где ключами являются слова, а значениями – списки соседних слов.
- **Очередь (deque):** Используется для реализации алгоритма поиска в ширину (BFS). Очередь хранит узлы, которые необходимо посетить.
- **Множество (set):** Используется для отслеживания посещенных узлов во время BFS, чтобы избежать заикливания.
- **Поиск в ширину (BFS):** Алгоритм для обхода графа, который гарантирует нахождение кратчайшего пути между двумя узлами (если граф не взвешенный).

Пример использования:

В примере использования создается граф для списка слов ["cab", "car", "cat", "bat", "bar", "mat"], а затем находится кратчайший путь от слова "cab" к слову "bat". Результат выводится на консоль.