

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Выполнил: Васюков Владислав Станиславович

Объяснение:

1. **Класс `Sight`:** Определяет структуру данных для представления достопримечательности с ее названием, ценностью и продолжительностью посещения. Метод `__repr__` используется для удобного вывода информации об объекте.
2. **Функция `greedy_sightseeing(sights, total_days)`:** Реализует жадную стратегию.
 - Сначала сортирует достопримечательности по убыванию их ценности (значимости).
 - Затем итерируется по отсортированному списку и выбирает достопримечательности, пока позволяет оставшееся время. Если достопримечательность помещается по времени, она добавляется в список `selected_sights`, а оставшееся время уменьшается.
3. **Функция `find_most_valuable_sight(sights, remaining_days)`:** Помогает в пошаговом жадном алгоритме. Она ищет в списке достопримечательностей ту, которая имеет наибольшую ценность и может быть посещена `за remaining_days`. Если таких достопримечательностей нет, возвращает `None`.
4. **Функция `greedy_sightseeing_step_by_step(sights, total_days)`:** Реализует жадный алгоритм, который на каждом шагу принимает решение о выборе достопримечательности, основываясь на локальной оптимальности (наибольшая ценность из доступных). Этот алгоритм более нагляден в плане того, как принимаются решения на каждом этапе.
 - В цикле `while True` на каждом шаге выбирается самая ценная достопримечательность, которую можно посетить (с помощью `find_most_valuable_sight`).
 - Если такая достопримечательность найдена, она добавляется в список `selected_sights`, время уменьшается, а достопримечательность удаляется из списка `available_sights` (чтобы избежать повторного выбора).
 - Цикл завершается, когда больше нет достопримечательностей, которые можно посетить.
 - Печатается информация о каждой выбранной достопримечательности и оставшемся времени.
5. **Пример использования:** Создается список достопримечательностей с их параметрами. Затем вызываются обе жадные функции для демонстрации их работы.
6. **Оптимальность:** Жадный алгоритм не гарантирует оптимальное решение. Например, может случиться так, что посещение двух достопримечательностей с меньшей ценностью даст в сумме большую ценность, чем посещение одной самой ценной, но из-за ограничений по времени жадный алгоритм сначала выберет самую ценную, а затем не сможет выбрать остальные.

Являются ли следующие алгоритмы жадными?

- **Сортировка пузырьком:** Нет. Сортировка пузырьком - это алгоритм сортировки, который не основан на жадной стратегии. Он сравнивает и меняет местами соседние элементы до тех пор, пока массив не будет полностью отсортирован.
- **Алгоритм Дейкстры для поиска кратчайшего пути:** Да. Алгоритм Дейкстры - это жадный алгоритм. Он на каждом шагу выбирает ближайший непосещенный узел, чтобы расширить путь, пока не будет найден кратчайший путь до целевого узла.
- **Алгоритм Краскала для построения минимального остовного дерева:** Да. Алгоритм Краскала - это жадный алгоритм. Он добавляет ребра в минимальный остовный лес в порядке возрастания их веса, если добавление ребра не создает цикл.
- **Алгоритм поиска в ширину (BFS):** Нет. Алгоритм поиска в ширину - это алгоритм обхода графа, который не является жадным. Он обходит граф по уровням, начиная с корневого узла.