Project. The megalopolis transport system

Условие задачи

Городская транспортная сеть представлена метро, автобусными маршрутами и железной дорогой. Эта сеть моделируется как неориентированный мультиграф: вершины — станции, рёбра — возможные перемещения между станциями по одному из видов транспорта. Между двумя станциями может существовать несколько рёбер, отличающихся видом транспорта и параметрами. Для каждого ребра задано базовое время в пути и текущая загруженность линии; фактическое время прохождения ребра определяется умножением базового времени на коэффициент, учитывающий загруженность и чувствительность соответствующего вида транспорта к перегрузкам. Кроме того, при смене вида транспорта на станции к общему времени маршрута добавляется штраф, который складывается из межвидового штрафа и локального времени пересадки на этой станции.

Необходимо, во-первых, исследовать связность транспортной сети. Для этого требуется выполнить поиск в глубину по четырём графам: по подграфу метро, по подграфу автобусов, по подграфу железной дороги и по объединённому графу (все рёбра). Для каждого из этих случаев нужно определить компоненты связности, выделить крупнейшую компоненту и назвать изолированными зонами все остальные компоненты. Результат

должен включать перечень таких изолированных зон, упорядоченный по невозрастанию размера компоненты; внутри компоненты станции перечисляются по возрастанию.

Во-вторых, для набора пассажирских запросов требуется рассчитать оптимальные маршруты. Каждый запрос задаёт начальную станцию, набор целевых станций и параметр, отвечающий за «цену» пересадки в итоговой оценке удобства. Критерий оптимальности для поиска пути — минимальное суммарное время в пути с учётом фактических времен рёбер и штрафов за пересадки при смене вида транспорта. Для корректного учёта штрафов рекомендуется рассматривать состояния вида «станция + последний использованный вид транспорта»; старт допускается без штрафа за первую посадку. Поиск кратчайших временных маршрутов необходимо проводить алгоритмом Дейкстры, где приоритет всегда у состояния с наименьшим известным временем; для выбора следующего состояния должна использоваться очередь с приоритетами.

Для каждой целевой станции нужно восстановить найденный по времени маршрут и оценить его удобство по метрике «время + коэффициент × число пересадок», где коэффициент берётся из запроса. Если до целевой станции существует несколько временно равных решений, следует предпочесть маршрут с меньшим числом пересадок. После этого для каждой группы целей внутри одного запроса требуется отсортировать полученные маршруты по возрастанию значения метрики удобства; при равенстве — по возрастанию общего времени, затем по возрастанию числа пересадок, затем по возрастанию номера целевой станции. Сортировку необходимо выполнить собственной реализацией алгоритма быстрой сортировки. В выводе для каждой целевой станции следует указать: пункт назначения, итоговое время, число пересадок, значение метрики удобства и явное описание пути через последовательность станций с пометкой, каким видом транспорта выполняется каждый переход. Если цель недостижима, это должно быть явно обозначено.

Таким образом, в задаче нужно: (1) с помощью поиска в глубину выявить изолированные зоны в каждом из четырёх рассматриваемых графов; (2) с помощью алгоритма Дейкстры и очереди с приоритетами рассчитать кратчайшие по времени маршруты с учётом перегруженности линий и штрафов за пересадки; (3) оценить удобство найденных маршрутов и

отсортировать их своей реализацией быстрой сортировки; (4) вывести упорядоченные результаты в требуемом формате.

Дополнительные условия и формальные детали задачи

Параметры модели

- Станции нумеруются от 1 до N.
- Вид транспорта mode ∈ {0,1,2} соответствует: метро, автобус, железная дорога.
- Фактическое время на ребре вычисляется как базовое время × (1 + доля загрузки × коэффициент чувствительности для данного вида). Доля загрузки — от 0 до 1 (или от 0% до 100%).
- Штраф за пересадку при смене вида транспорта на станции складывается из межвидового штрафа (матрица 3×3) и локального времени пересадки на этой станции. Для первой посадки штраф не взимается.

Требуемые алгоритмы и структуры данных

- **DFS (поиск в глубину)** — для нахождения компонент связности: отдельно по метро, автобусам, железной дороге и по объединённому графу.
- **Алгоритм Дейкстры** — для кратчайшего по времени пути при состоянии «станция + последний вид транспорта», с возможностью восстановления маршрута через хранение предков.
- **Очередь с приоритетами** — для выбора состояния с минимальным текущим временем в реализации Дейкстры.
- **Быстрая сортировка** — собственная реализация для упорядочивания списка маршрутов по метрике удобства с описанными тай-брейками.

Требования к реализации

- Граф хранить в виде списков смежности; поддерживать несколько рёбер между одной парой станций.

- В Дейкстре использовать очередь с приоритетами с приоритетом у состояния с минимальным временем; состояние — «станция + последний вид транспорта».
- При переходе между рёбрами с разными видами транспорта добавлять соответствующий межвидовой штраф и время пересадки на текущей станции; для первой посадки штраф не начисляется.
- При равных временах до одной цели выбирать маршрут с меньшим числом пересадок (это можно учитывать вторым ключом в расстоянии либо при восстановлении).
- Для сортировки маршрутов по метрике удобства использовать **собственную** реализацию быстрой сортировки (без стандартных сортировщиков).
- В выводе строго придерживаться требуемого формата и порядка сортировки.

Методическое планирование

План подготовки к реализации программного кода проекта:

1. Основы алгоритмов и их анализ (1.1, 1.2, 2.2, 3.1, 3.2)
2. Принцип разделяй и властвуй (4.1, 4.4, 4.5, 4.6)
3. Пирамиды и очередь с приоритетом (10.1, 10.3, 6.1, 6.2, 6.3, 6.4, 6.5)
4. Элементарные алгоритмы на графах (22.1, 22.2, 22.3, 22.4, 3.1, 3.2, 10.1, 10.2)
5. Алгоритм кратчайших путей (Алгоритм Дейкстры) (22.1, 22.3, 6.5, 10.1, 10.2, 10.3, 24.1, 24.3, 24.4, 24.5)
6. Быстрая сортировка (7.1, 7.2, 7.3, 7.4)

Планируемое время подготовки к реализации проекта: 4 недели, на каждой недели 3 дня, в каждый из дней по 3-4 часа

Неделя 1. Основы алгоритмов и их анализ

День 1. (3-4 часа)

Тема: Введение в алгоритмы

- Гл. 1.1–1.2 — Что такое алгоритмы, алгоритмы как технология
- Гл. 2.2 — Анализ алгоритмов
- Гл. 3.1 — Асимптотические обозначения (O, Ω, Θ)

Практика: рассчитать сложность вставочной сортировки.

День 2. (3-4 часа)

Тема: Принцип «Разделяй и властвуй»

- Гл. 4.1–4.5 — рекурсия, подстановка, деревья рекурсии
- Разбор примеров: бинарный поиск, сортировка слиянием

Пратика: написать простую рекурсивную сортировку

День 3. (3-4 часа)

Тема: Практика анализа сложности

- Гл. 3.2 — стандартные функции роста
- Гл. 4.6 — доказательство основной теоремы

Практика: проанализировать сложность сортировки слиянием и бинарного поиска.

Неделя 2. Быстрая сортировка

День 4. (3-4 часа)

Тема: Теория быстрой сортировки

- Гл. 7.1–7.2 — описание и анализ производительности

Практика: реализовать простую быструю сортировку, замерить время работы.

День 5. (3-4 часа)

Тема: Рандомизированная быстрая сортировка

- Гл. 7.3–7.4 — случайный выбор опорного элемента

Практика: написать версию с рандомизацией и сравнить с детерминированной.

День 6. (3-4 часа)

Тема: Сравнение алгоритмов сортировки

- Гл. 6.1–6.4 (пирамидальная сортировка) — обзор для сравнения подходов

Практика: провести сравнение quicksort и heapsort на массивах разного размера.

Неделя 3. Очередь с приоритетами и основы графов

День 7 (3-4 часа)

Тема: Пирамиды

- Гл. 6.1–6.3 — структура и построение пирамид

Практика: реализовать heapify() и построение кучи из массива.

День 8. (3-4 часа)

Тема: Очереди с приоритетами

- Гл. 6.5 — операции INSERT, EXTRACT-MAX, INCREASE-KEY

Практика: реализовать очередь с приоритетами и протестировать на примере задачи о расписании.

День 9. (3-4 часа)

Тема: Основы графов

- Гл. 22.1 — представление графов (списки смежности, матрицы)
- Гл. 22.2 — поиск в ширину (BFS) как база для DFS

Практика: написать реализацию графа и BFS.

Неделя 4. Поиск в глубину и Алгоритм Дейкстры

День 10 (3-4 часа)

Тема: Поиск в глубину (DFS)

- Гл. 22.3 — DFS, рекурсивная и итеративная реализация

Практика:

- Реализовать DFS на графе
- Найти сильно связанные компоненты.

День 11 (3-4 часа)

Тема: Кратчайшие пути

- Гл. 24.1–24.3 — Беллман-Форд и Дейкстра

Практика: реализовать алгоритм Дейкстры с очередью с приоритетами.

День 12 (3-4 часа)

Тема: Закрепление

Повтор ключевых идей (анализ, структуры данных, графы)

Практика:

- Написать мини-проект:

  “Поиск кратчайших путей между городами” или

  “Планировщик задач с приоритетами”.
- Повторить анализ сложности и структуру кода.