Лабораторная работа № 7 по курсу дискретного анализа: Жадные алгоритмы

Выполнил студент группы 08-303 МАИ *Арусланов Кирилл.*

Условие

Даны N отрезков $[L_i, R_i]$ на координатной прямой. Необходимо выбрать минимальное количество отрезков, которые полностью покрывают интервал $[0, M]$.

Метод решения

Для решения задачи применяется **жадный алгоритм**. Пусть текущая правая граница покрытия равна `cur`. Среди всех отрезков, начинающихся не правее `cur`, выбирается тот, который имеет наибольший конец `R`. Этот отрезок добавляется в решение, и правая граница `cur` сдвигается до `R`. Процесс повторяется, пока `cur < M`. Если на каком-то шаге подходящих отрезков не находится, покрытие невозможно.

Сложность алгоритма:

$$O(N \log N)$$

— на сортировку по левым концам,

$$O(N)$$

— на один проход по массиву. Итого — асимптотика $O(N \log N)$.

Описание программы

Программа реализована на языке C++.

- **struct Seg** — структура для хранения отрезка (L, R) и исходного индекса.
- Основной алгоритм:
  1. Чтение N и всех отрезков.
  2. Сортировка отрезков по L с помощью лямбда-функции.
  3. Итеративный жадный выбор отрезков, расширяющих покрытие.
  4. Вывод результата в исходном порядке появления.

Дневник отладки

Проблем при разработке не возникало, программа прошла чеккер с первой попытки.

Тест производительности

Были проведены эксперименты на наборах данных размером $N = 1000$, 5000, 10000, 20000, 50000. В таблице приведены усреднённые времена выполнения по 5 испытаниям (в миллисекундах):

N	M	Среднее время работы, мс
1000	1000	0.08
5000	5000	0.43
10000	10000	0.79
20000	20000	2.7
50000	50000	10.9

Рост времени работы соответствует зависимости $O(N \log N)$: при увеличении N в 5 раз время увеличивается примерно в 10–12 раз, что согласуется с теоретической сложностью сортировки.

Недочёты

Программа предполагает, что входные данные корректны и не содержит проверки на отрицательные значения координат или пустые отрезки.

Выводы

Реализован жадный алгоритм минимального покрытия интервала. Подход эффективен и оптимален для данной задачи: на каждом шаге выбирается локально лучший отрезок, что приводит к глобально оптимальному решению. Алгоритм имеет временную сложность $O(N \log N)$ и хорошо масштабируется при увеличении объёма входных данных.

Данный метод применяется в задачах оптимального планирования, интервалов времени, покрытия диапазонов, маршрутизации и обработке событий.