

**Московский авиационный институт
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной
математики**

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: Т. Д. Голубев
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б-22
Дата:
Оценка:
Подпись:

Москва, 2024

Лабораторная работа №8

Задача: На координатной прямой даны несколько отрезков с координатами $[L_i, R_i]$. Необходимо выбрать минимальное количество отрезков, которые бы полностью покрыли интервал $[0, M]$.

1 Описание

Требуется решить задачу с помощью жадных алгоритмов.

В [1] сказано: «Такой алгоритм делает на каждом шаге локально оптимальный выбор, – в надежде, что итоговое решение также окажется оптимальным».

Я предлагаю отсортировать исходный массив отрезков по левой и правой границе. Затем итерироваться по нему и на каждом шаге искать самый большой отрезок, который покрывает ещё непокрытую на предыдущих этапах часть. Соответственно, на каждом шаге нужно обновлять границы непокрытого исходного отрезка.

Сложность такого алгоритма – $O(n \cdot \log(n))$, где n – количество отрезков. Такая сложность получается, потому что я использую quick sort.

2 Исходный код

main.cpp	
bool IndexComparator(const Segment& a, const Segment& b)	Компаратор для сортировки по индексу.
bool SegmentComparator(const Segment& a, const Segment& b)	Компаратор для сортировки по левой и правой границе.

```
1 struct Segment {
2     int64_t start;
3     int64_t end;
4     size_t index;
5
6     Segment() = default;
7
8     Segment(int64_t start, int64_t end, size_t index) : start{start}, end{end}, index{
9         index} {}
10
11     int64_t ComputeCover(int64_t left, int64_t m) {
12         if (end < 0 || start > m) {
13             return 0;
14         }
15         return std::min<int64_t>(m, end) - std::max<int64_t>(left, start);
16     };
17 }
```

3 Консоль

```
cat-mood@nuclear-box:~/programming/mai-da-labs/lab08/build$ ./lab08_exe
6
1 5
0 1
1 2
2 3
3 4
4 5
5
2
1 5
0 1
```

4 Тест производительности

Тест производительности представляет из себя следующее: Решение с использованием жадных алгоритмов сравнивается с наивным обходом, где для каждого шага покрытия перебираются все сегменты N , а общее число шагов ограничено длиной интервала M . Количество отрезков в первом тесте – 6, во втором – 30.

```
cat-mood@nuclear-box:~/programming/mai-da-labs/lab07/build$ ../a.out <test1.txt
Naive: 27 ms
My solution: 10 ms
```

```
cat-mood@nuclear-box:~/programming/mai-da-labs/lab08/build$ ../a.out <test2.txt
Naive: 39 ms
My solution: 14 ms
```

Как видно, моё решение работает быстрее.

5 Выводы

Выполнив восьмую лабораторную работу по курсу «Дискретный анализ», я решил задачу с помощью жадных алгоритмов. В ходе работы столкнулся с проблемой поиска оптимального решения. Изначально я выбирал отрезок, который максимально покрывает весь интервал, но позже я понял, что такой отрезок должен покрывать начало исходного интервала.

Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн
Алгоритмы: построение и анализ. - 3-е изд. - М.: ООО "И. Д. Вильямс 2013. -
1328 с.