

Раскраска графов и покрытия

Лабораторная работа №3

Осенний семестр, 2024 г.

1. Определить, является ли граф совершенным:

- Постройте произвольный граф с помощью функции `Graph` (используйте не менее 6 вершин и 9 рёбер);
- Проверьте, является ли граф совершенным, используя хроматическое число и кликовое число его подграфов (наименьшее количество цветов, необходимых для раскраски любого подграфа и максимальный размер клики этого подграфа должны быть равны для совершенных графов);
- Для проверки используйте функции `VertexChromaticNumber` и `FindClique`.
- Также можно воспользоваться функцией <https://resources.wolframcloud.com/FunctionRepository/resources/PerfectGraphQ>

2. Найти максимальное паросочетание и независимое множество в графе:

- Создайте двудольный граф с не менее чем 8 вершинами и 12 рёбрами;
- Используйте функцию `FindIndependentEdgeSet` для нахождения максимального паросочетания;
- Используйте функцию `FindIndependentVertexSet` для нахождения независимого множества;
- Визуализируйте исходный граф, максимальное паросочетание и независимое множество на нём, выделив рёбра паросочетания и вершины независимого множества другим цветом.

3. Исследовать граф и найти минимальные вершинное и рёберное покрытия:

- Постройте "`DodecahedralGraph`" с помощью встроенной функции `GraphData`;
- Найдите минимальное вершинное покрытие графа, используя функцию `FindVertexCover`;
- Найдите минимальное рёберное покрытие графа с помощью функции `FindEdgeCover`;
- Визуализируйте граф и выделите вершины вершинного покрытия и рёбра рёберного покрытия разными цветами.

4. RLE-сжатие — один из самых простых методов сжатия строки, основанный на сокращении подстрок, состоящих из одинаковых символов. Сжатие осуществляется следующим образом:

- Строка разбивается на минимальное количество подстрок, состоящих из одинаковых символов. Например, `abbcaaa` превращается в строки `a`, `bb`, `c`, `aaa`.
- Каждая из полученных строк превращается в строку, состоящую из числа и буквы. Числом является количество повторений символа в этой строке, буква берётся из первого символа обрабатываемой строки. Число не добавляется, если количество символов в строке равно единице. Из предыдущего массива строк мы получаем `a`, `2b`, `c`, `3a`.

- Затем полученные строки конкатенируются в исходном порядке. В рассмотренном примере в итоге получим a2bc3a.

Вам дана строка s , уже сжатая в RLE-формате. Назовём строку, из которой была получена s , строкой t . Вам даны q запросов, каждый из них представлен целыми числами l и r . В каждом запросе вам необходимо найти длину сжатой подстроки $t[l \dots r]$.

Формат ввода

В первой строке входного файла записана строка s , состоящая из строчных букв латинского алфавита и цифр ($1 \leq |s| \leq 1000000$). Гарантируется, что существует такая непустая строка t , из которой RLE-сжатием получается строка s . Также гарантируется, что в строке t не было больше 1000000000 одинаковых подряд идущих символов.

В следующей строке дано количество запросов q ($1 \leq q \leq 100000$). Каждая из следующих q строк содержит два числа l_i и r_i ($1 \leq l_i \leq r_i \leq |t|$) — параметры запросов.

Формат вывода

Выведите q чисел, каждое в отдельной строке — ответы на запросы в том порядке, в котором запросы были заданы во входных данных.

Пример 1

Ввод

a2bc3a

5

1 7

5 7

1 2

3 5

4 4

Вывод

6

2

2

3

1

Пример 2

Ввод

x1000000000yz

3

2 1000000001

2 1000000002

5938493 15938493

Вывод

11

12

9