

Рациональные числа

Ratio – отношение, дробь

$$r = \frac{m}{n}; \quad m \in \mathbb{Z} \text{ (целое)}, \quad n \in \mathbb{N} \text{ (натуральное)}$$

Сложение и умножение

$$\frac{m_1}{n_1} + \frac{m_2}{n_2} = \frac{m_1 n_2 + m_2 n_1}{n_1 n_2}, \quad \frac{m_1}{n_1} \times \frac{m_2}{n_2} = \frac{m_1 m_2}{n_1 n_2}.$$

Алгоритм Евклида нахождения Наибольшего Общего Делителя НОД(a, b)

- для двух ненулевых целых чисел a и b , цикл пока $b \neq 0$:

$$t \leftarrow b$$

$$b \leftarrow a \bmod b$$

$$a \leftarrow t$$

- по завершению цикла a – искомый НОД

Задание I: запрограммировать рациональные числа как класс:

1 Конструкторы:

- инициализация числителем (m) и знаменателем (n)
- инициализация целым числом
- конструктор по умолчанию

2 Оператор вывода рационального числа на печать

3 Операторы унарные: $-$ и $++$, $--$ (префиксные и постфиксные); и бинарные: $+$, $-$, $*$, $/$, $+=$, $-=$, $*=$, $/=$

4 Функцию возведения в целую степень: `pow()`

5 Преобразования `rational` \iff `double`:

- конструктор-преобразование из `double` с заданной точностью
- оператор преобразования рационального числа в `double`

- предусмотрите преобразование к несократимому виду
- для всех операций и функций должны быть тестовые примеры
- в пятом пункте возможно использовать `explicit`

Использование класса рациональных чисел

Вычисление чисел Бернулли: алгоритм Akiyama-Tanigawa

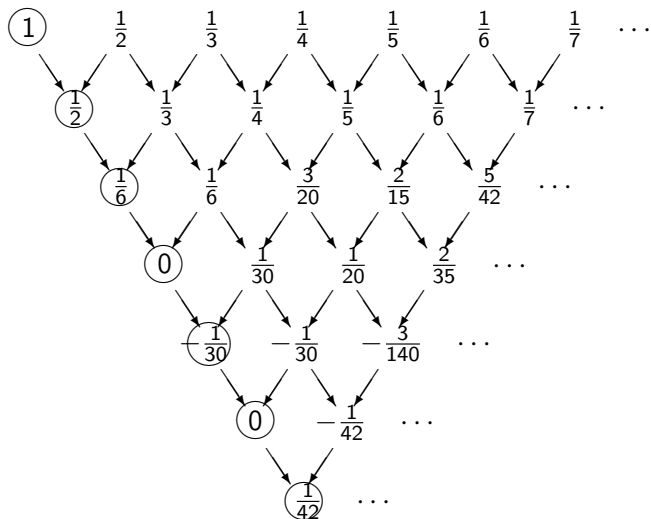
- Задаётся целое число n
- Цикл по i от 0 до n с шагом 1:
$$A[i] \leftarrow \frac{1}{i+1}$$

Цикл по j от i до 1 с шагом -1 :
$$A[j-1] \leftarrow j \times (A[j-1] - A[j])$$
- В конце получаем $A[0]$, которое и есть B_n
- На следующем слайде графическая иллюстрация алгоритма и значения B_{1-16}

Задание II

- Написать функцию возвращающую N первых чисел Бернулли: (возвращающую `vector<Rational>`)
- Вычислить B_{2k} , $k = 1 \dots 8$

Треугольник Akiyama-Tanigawa



$$B_0 = 1;$$

$$B_1 = 1/2;$$

$$B_2 = 1/6;$$

$$B_4 = -1/30;$$

$$B_6 = 1/42;$$

$$B_8 = -1/30;$$

$$B_{10} = 5/66;$$

$$B_{12} = -691/2730;$$

$$B_{14} = 7/6;$$

$$B_{16} = -3617/510$$