

5.20. Реализуйте проверку числа N на простоту в compile-time с асимптотикой $O(\sqrt{N})$ с помощью шаблонной рекурсии (без использования constexpr, а также математических функций стандартной библиотеки).

Все что нам нужно - это реализовать метафункцию для поиска квадратного корня и подставить его в IsPrimeHelper из билета 4.11. вместо $N-1$ (так как нам достаточно проверить только делимость на числа $1 \dots \sqrt{N}$). Корень находим бинарным поиском за логарифм, значит итоговая асимптотика $O(\log N + \sqrt{N}) = O(\sqrt{N})$

```
1 template<size_t N, size_t L0=1, size_t HI=N>
2 struct Sqrt { // используем бинарный поиск
3 private:
4     // вычисляем середину округленную вверх
5     static const size_t mid = (L0 + HI + 1) / 2;
6 public:
7     // сравниваем N с квадратом середины интервала
8     static const size_t value = (N < mid * mid) ? Sqrt<N, L0, mid - 1>::value
9                                         : Sqrt<N, mid, HI>::value;
10 };
11
12 // частичная специализация для случая L0 = HI (нашли ответ)
13 template<size_t N, size_t M>
14 struct Sqrt<N, M, M> {
15     static const size_t value = M;
16 };
17
18 template <size_t N>
19 struct IsPrime { // просто подставляем корень из N в IsPrimeHelper
20     static const bool value = IsPrimeHelper<N, Sqrt<N>::value>::value;
21 };
22
23 template <>
24 struct IsPrime<1> { // отдельно случай для 1
25     static const bool value = false;
26 };
```