Modern C++

2. Task: sqrt

30.10.2017

Sidnev A.A.

Sqrt(x). Version 1

```cpp
class Solution {
  vector<int> sqr;
public:
  Solution() : sqr(46341) {
    iota(sqr.begin(), sqr.end(), 0);
  }
  int mySqrt(int x) {
    if (x < 0) {
      return INT_MIN;
    }
    auto root = upper_bound(sqr.begin(), sqr.end(), x,
                            [](const int& x, const int& i) {
      return x < i * i;
    });
    return *(--root);
  }
};
```

Sqrt(x). Version 2

```cpp
class Solution {
  vector<int> init() {
    vector<int> sqr(46341);
    iota(sqr.begin(), sqr.end(), 0);
    return sqr;
  }

public:
  int mySqrt(int x) {
    static vector<int> sqr = init();

    if (x < 0) {
      return INT_MIN;
    }

    auto root = upper_bound(sqr.begin(), sqr.end(), x, [](const int& x, const int& i) {
      return x < i * i;
    });

    return *(--root);
  }
};
```

Sqrt(x). Version 3

```cpp
class Solution {
  array<int, 46341> init() {
    array<int, 46341> sqr;
    iota(sqr.begin(), sqr.end(), 0);
    return sqr;
  }

public:
  int mySqrt(int x) {
    static array<int, 46341> sqr = init();

    if (x < 0) {
      return INT_MIN;
    }

    auto root = upper_bound(sqr.begin(), sqr.end(), x, [](const int& x, const int& i) {
      return x < i * i;
    });

    return *(--root);
  }
};
```

Sqrt(x). Version 4

```cpp
class Solution {
public:
  int mySqrt(int x) {
    if (x < 0) {
      return INT_MIN;
    }
    else {
      int l = 0;
      int r = 46340;

      while (l < r) {
        int m = l + (r - l) / 2,
            p = m * m;
        if (p == x) {
          return m;
        } else if (p < x) {
          l = m + 1;
        } else {
          r = m;
        }
      }
                                                    if (l * l > x) {
                                                          return l - 1;
                                                    } else {
                                                          return l;
                                                    }
                                                }
                                            }
                                        };
```