LOBACHEVSKY
UNIVERSITY

Advanced C++

constexpr example

Sidnev A.A.

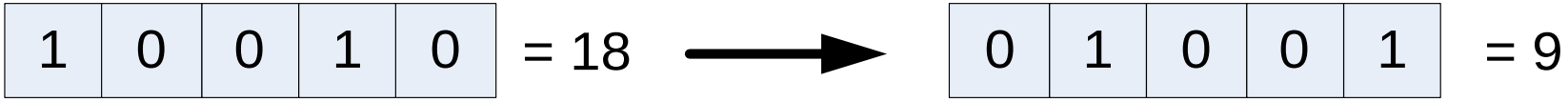
515. Find Largest Value in Each Tree Row

LOBACHEVSKY UNIVERSITY

```
class Solution {
public:
  vector<int> largestValues(TreeNode* root) {
    vector<int> out;
    if (root == nullptr)
      return out;

    queue<TreeNode*> q;
    q.emplace(root);
    while (!q.empty()) {
      int size = q.size();
      int max_val = q.front()->val;
      for (int i = 0; i < size; ++i) {
        auto node = q.front();
        q.pop();
        max_val = max(max_val, node->val);

        if (node->left)
          q.push(node->left);
        if (node->right)
          q.push(node->right);
      }
      out.emplace_back(max_val);
    }
    return out;
  }
};
```

Reverse bits

Task: reverse bits



0 0 0 0 = 0	→	0 0 0 0 = 0		1 0 0 0 = 8	→	0 0 0 1 = 1
0 0 0 1 = 1	→	1 0 0 0 = 8		1 0 0 1 = 9	→	1 0 0 1 = 9
0 0 1 0 = 2	→	0 1 0 0 = 4		1 0 1 0 = 10	→	0 1 0 1 = 5
0 0 1 1 = 3	→	1 1 0 0 = 12		1 0 1 1 = 11	→	1 1 0 1 = 13
0 1 0 0 = 4	→	0 0 1 0 = 2		1 1 0 0 = 12	→	0 0 1 1 = 3
0 1 0 1 = 5	→	1 0 1 0 = 10		1 1 0 1 = 13	→	1 0 1 1 = 11
0 1 1 0 = 6	→	0 1 1 0 = 6		1 1 1 0 = 14	→	0 1 1 1 = 7
0 1 1 1 = 7	→	1 1 1 0 = 14		1 1 1 1 = 15	→	1 1 1 1 = 15

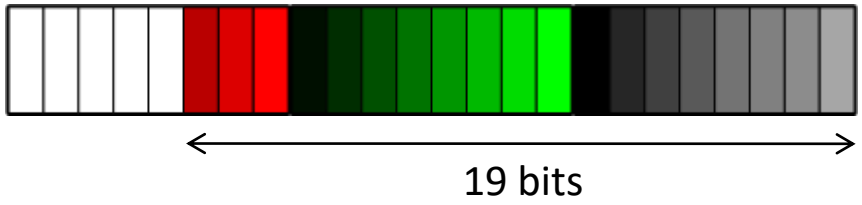
Reverse bits: implementation

LOBACHEVSKY UNIVERSITY

```
int reverse(int val, int bits_count) {
  int mask = 1 << (bits_count - 1);
  int rev_val = 0;
  for (int i = 0; i < bits_count; i++) {
    int bit = val & mask ? 1 : 0;
    rev_val |= bit << i;
    mask = mask >> 1;
  }
  return rev_val;
}

int main() {
  for (int i = 0; i < 16; ++i) {
    std::cout << i << " -> " << reverse(i, 4) << std::endl;
  }
}
```

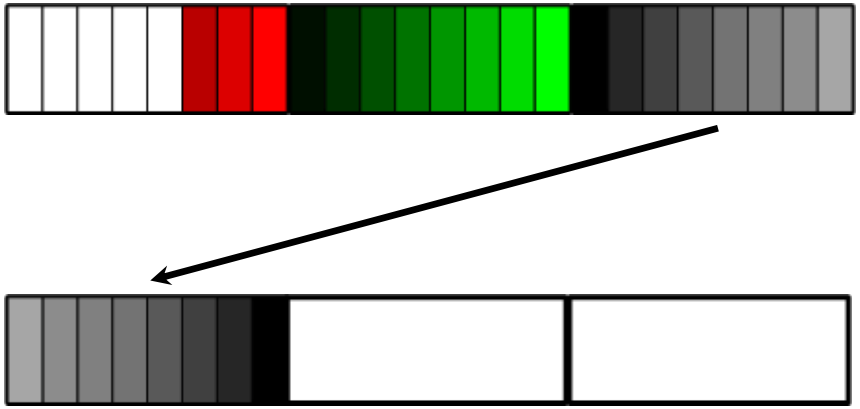
Reverse bits: lookup table (1)

- Reverse 19-bits length number



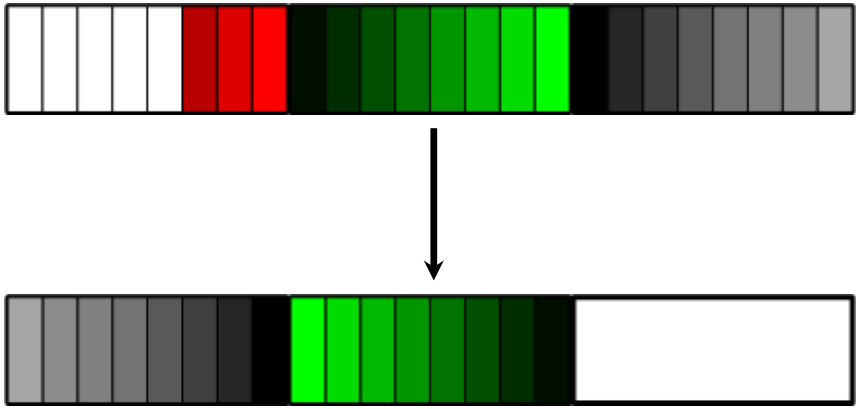
Reverse bits: lookup table (2)

- Write first byte



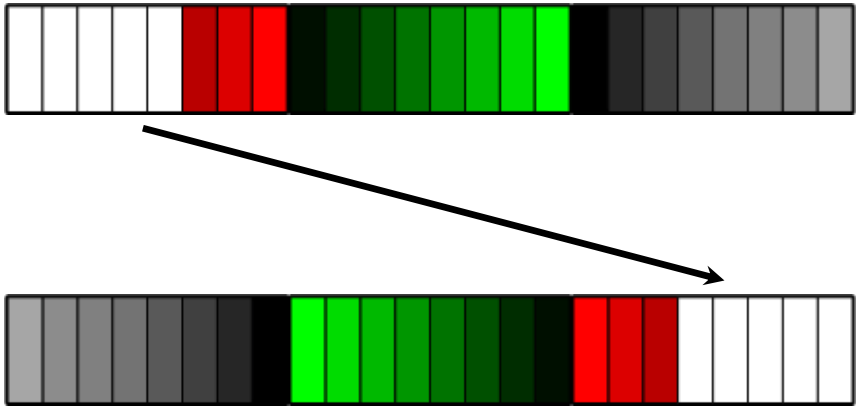
Reverse bits: lookup table (3)

- Write second byte



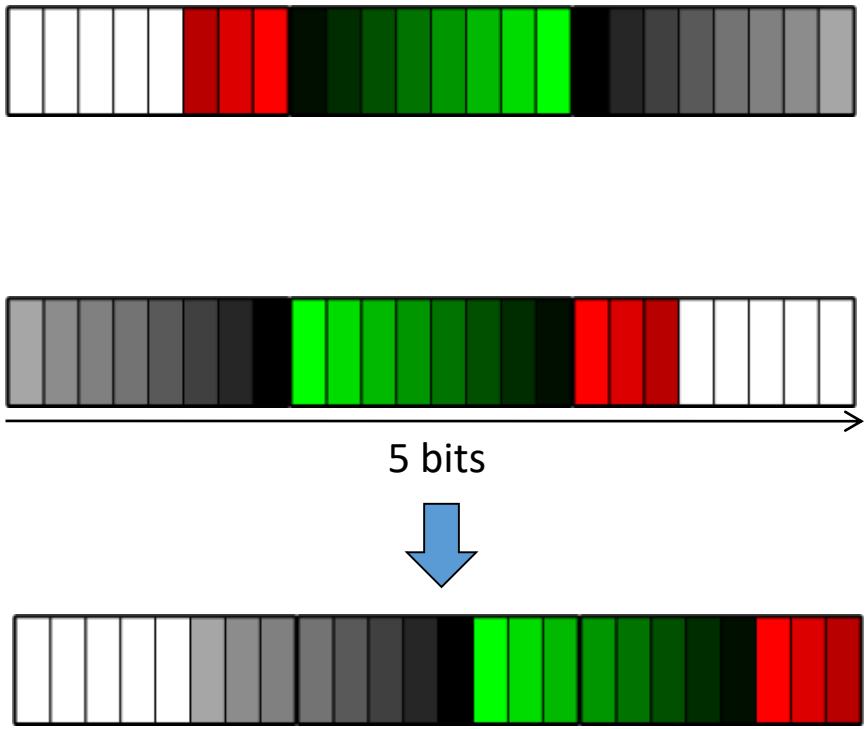
Reverse bits: lookup table (4)

- Write third byte



Reverse bits: lookup table (5)

- Shift 5 bits to the right



Without constexpr

LOBACHEVSKY UNIVERSITY

```
unsigned char rev[256]={
0, 128, 64, 192, 32, 160, 96, 224, 16, 144, 80, 208, 48, 176, 112, 240, 8, 136, 72,
200, 40, 168, 104, 232, 24, 152, 88, 216, 56, 184, 120, 248, 4, 132, 68, 196, 36,
164, 100, 228, 20, 148, 84, 212, 52, 180, 116, 244, 12, 140, 76, 204, 44,  172, 108,
236, 28, 156, 92, 220, 60, 188, 124, 252, 2, 130, 66, 194, 34, 162, 98, 226, 18, 146,
82, 210, 50, 178, 114,  242, 10, 138, 74, 202, 42, 170, 106, 234, 26, 154, 90, 218,
58, 186, 122, 250, 6, 134, 70, 198, 38, 166, 102, 230, 22,  150, 86, 214, 54, 182,
118, 246, 14, 142, 78, 206, 46, 174, 110, 238, 30, 158, 94, 222, 62, 190, 126, 254,
1, 129, 65, 193, 33, 161, 97, 225, 17, 145, 81, 209, 49, 177, 113, 241, 9, 137, 73,
201, 41, 169, 105, 233, 25, 153, 89, 217, 57, 185, 121, 249, 5, 133, 69, 197, 37,
165, 101, 229, 21, 149, 85, 213, 53, 181, 117, 245, 13, 141, 77, 205, 45, 173, 109,
237, 29, 157, 93, 221, 61, 189, 125, 253, 3, 131, 67, 195, 35, 163, 99, 227, 19, 147,
83, 211, 51, 179, 115, 243, 11, 139, 75, 203, 43, 171, 107, 235, 27, 155, 91, 219,
59, 187, 123, 251, 7, 135, 71, 199, 39, 167, 103, 231, 23, 151, 87, 215, 55, 183,
119, 247, 15, 143, 79, 207, 47, 175, 111, 239, 31, 159, 95, 223, 63, 191, 127, 255};
```

Reverse bits: constexpr (1)

LOBACHEVSKY UNIVERSITY

```
constexpr int reverse(int val, int bits_count) {
  int mask = 1 << (bits_count - 1);
  int rev_val = 0;
  for (int i = 0; i < bits_count; i++) {
    int bit = val & mask ? 1 : 0;
    rev_val |= bit << i;
    mask = mask >> 1;
  }
  return rev_val;
}

const int bits_count = 5;

constexpr int pow2() {
  int val = 1 << bits_count;
  return val;
}
```

Reverse bits: constexpr (2)

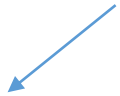
LOBACHEVSKY UNIVERSITY

```
constexpr auto makeTable() {
  constexpr int size = pow2();
  std::array<int, size> table = { 0 };
  for (int i = 0; i < size; ++i) {
    table[i] = reverse(i, bits_count);
  }
  return table;
}

constexpr auto reverseTable = makeTable();

int main() {
  for (int i = 0; i < pow2(); ++i) {
    std::cout << i << " -> " << reverseTable[i] << std::endl;
  }
}
```

Reverse bits: constexpr (3)

LOBACHEVSKY UNIVERSITY

```
template<int bits_count>
constexpr auto makeTable() {
  constexpr int size = pow2(bits_count);
  std::array<int, pow2(bits_count)> table = { 0 };
  for (int i = 0; i < size; ++i) {
    table[i] = reverse(i, bits_count);
  }
  return table;
}

constexpr int kReverseTableBits = 5;
constexpr auto reverseTable = makeTable<kReverseTableBits>();
int main() {
  for (int i = 0; i < pow2(kReverseTableBits); ++i) {
    std::cout << i << " -> " << reverseTable[i] << std::endl;
  }
}
```



std::array is literal type

std::vector is **not** literal type