**QR Code Tutorial**

**Data Masking**

Now that the modules have been placed in the matrix, the best mask pattern must be determined. A mask pattern changes which modules are dark and which are light according to a particular rule. The purpose of this step is to modify the QR code to make it as easy for a QR code reader to scan as possible.

**Terminology: Masking**

If a module in the QR code is "masked", this simply means that if it is a light module, it should be changed to a dark module, and if it is a dark module, it should be changed to a light module. In other words, masking simply means to toggle the color of the module.

**Overview of Mask Patterns**

The QR code specification defines eight mask patterns that can be applied to the QR code. For example, for mask pattern #1, every even-numbered row in the QR matrix is masked, and for mask pattern #2, every third column in the QR matrix is masked.

Please refer to the mask patterns page for more details about the eight mask patterns.

**What to Mask**

Mask patterns must ONLY be applied to data modules and error correction modules. In other words:

* Do not mask function patterns (finder patterns, timing patterns, separators, alignment patterns)
* Do not mask reserved areas (format information area, version information area)

**Determining the Best Mask**

After a mask pattern has been applied to the QR matrix, it is given a penalty score based on four evaluation conditions that are defined in the QR code specification. A QR code encoder must apply all eight mask patterns and evaluate each one. Whichever mask pattern results in the lowest penalty score is the mask pattern that must be used for the final output.

**How to Evaluate Reserved Areas**

Note that the entire matrix (including function patterns and reserved areas) is evaluated, even though the masking is only applied to data and error correction modules.

The Four Penalty Rules

**The four penalty rules can be summarized as follows:**

* The first rule gives the QR code a penalty for each group of five or more same-colored modules in a row (or column).
* The second rule gives the QR code a penalty for each 2x2 area of same-colored modules in the matrix.
* The third rule gives the QR code a large penalty if there are patterns that look similar to the finder patterns.
* The fourth rule gives the QR code a penalty if more than half of the modules are dark or light, with a larger penalty for a larger difference.

**QR Mask Patterns Explained**

When encoding a QR code, there are eight mask patterns that you can use to change the outputted matrix. Each mask pattern changes the bits according to their coordinates in the QR matrix. The purpose of a mask pattern is to make the QR code easier for a QR scanner to read.

The Mask Patterns

Each mask pattern uses a formula to determine whether or not to change the color of the current bit. You put the coordinates of the current bit into the formula, and if the result is 0, you use the opposite bit at that coordinate. For example, if the bit for coordinate (0,3) is 1, and the formula is equal to 0 for that coordinate, then you put a 0 at (0,3) instead of a 1.

Here is the list of the mask pattern formulas. Note that some versions of the QR code standard have had errors in the section about mask patterns. The information below has been corrected.

|  |  |
| --- | --- |
| **Mask Number** | **If the formula below is true for a given row/column coordinate, switch the bit at that coordinate** |
|  |  |
| 0 | (row + column) mod 2 == 0 |
| 1 | (row) mod 2 == 0 |
| 2 | (column) mod 3 == 0 |
| 3 | (row + column) mod 3 == 0 |
| 4 | ( floor(row / 2) + floor(column / 3) ) mod 2 == 0 |
| 5 | ((row \* column) mod 2) + ((row \* column) mod 3) == 0 |
| 6 | ( ((row \* column) mod 2) + ((row \* column) mod 3) ) mod 2 == 0 |
| 7 | ( ((row + column) mod 2) + ((row \* column) mod 3) ) mod 2 == 0 |

