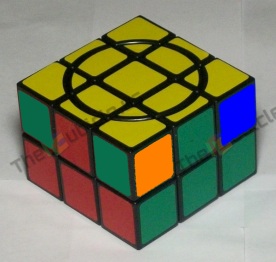
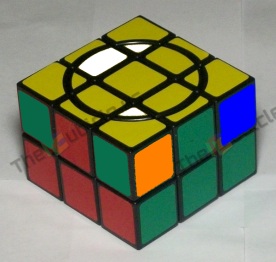
Proponuję wprowadzić następujące oznaczenia dla części:

* CP (Corners Permutation) – permutacja zewnętrznych rogów
* EP (Edges Permutation) – permutacja zewnętrznych krawędzi
* cp – permutacja wewnętrznych rogów
* ep – permutacja wewnętrznych krawędzi
* C - centry

Oraz następujące oznaczenia numerów klocków, oraz slotów w których się znajdują. Pamiętać będziemy numery klocków jakie znajdują się w danym slocie, poniżej przykłady.

* CP – sloty [0, 6] (7 jest stałym klockiem, nie trzeba go pamiętać), przechowujemy wartości [0, 6]
* EP – sloty [0, 7], przechowujemy wartości [0, 6]
* cp – sloty [0, 6] (7 jest stałym klockiem), wystarczy przechowywać 0 lub 1 na każdej pozycji (czy klocek dobrego koloru – 0, złego - 1)
* ep – sloty [0,6], jak wyżej 0 lub 1 na danej pozycji
* C – flaga [0, 1] (0 – dobry, 1 - zły)

Przykłady

CP (dla ułożonej kostki)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Numer | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

CP (dla kości po prawej)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Numer | 3 | 0 | 2 | 1 | 4 | 5 | 6 |

CP (dla kości po prawej)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Numer | 3 | 0 | 2 | 1 | 4 | 5 | 6 |

EP (dla kości po prawej)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Numer | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

cp (dla kości po prawej)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Numer | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

ep (dla kości po prawej)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Numer | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

C = 0

Całą kostkę można w ten sposób przechować w czterech tablicach oraz jednej fladze. Indeksy elementów takich tablic byłyby numerami slotów, natomiast wartości w komórkach numerami klocków.

Zamiast tablic posłużymy się jednak poszczególnymi bitami liczb całkowitych. Ciągi reprezentujące CP, EP, cp, ep będą rozłożone na kolejnych bitach.

Jak widać, cp i ep można przedstawić jako ciąg piętnastu 0 i 1. C to jest jeden bit. Czyli można je przechować na 16 bitach.

EP to osiem liczb, które można zakodować na trzech bitach każda (przedział [0, 7], 7(dec) = 111(bin)). Czyli razem 8 x 3 bity = 24 bity.

Analogicznie CP to siedem liczb. Stąd 7 x 3 bity = 21 bitów

Wszystko razem można przechować na 16b + 24b +21b = 61 bitach, czyli zmienna typu long long int. To będzie reprezentacja naszej kości.

Wykonanie ruchu, np U przemieszcza klocki. W naszym przypadku będzie to zamiana miejscami bitów, na których dane klocki się znajdują.

Wykonanie każdego ruchu (U, U’, U2, D, D’, D2, R2, F2, L2, B2 oraz te specjalne) można zaimplementować jako funkcje, które manipulują na bitach liczby, która reprezentuje stan kostki.

Na początek, zanim zaczniemy klepać kod trzeba się jeszcze zająć następującymi rzeczami:

* Przemyśleć co na których bitach najlepiej będzie przechować (po kątem optymalizacji ruchów – im mniej operacji bitowych będzie wykonywać funkcja wykonująca ruch tym lepiej )
* Opracować wykonywanie ruchów – opisać jakie bity i w jaki sposób będzie modyfikować każdy ruch
* Jak już będzie wiadomo co i jak modyfikuje dany ruch, można się zastanowić jakich operacji bitowych trzeba użyć żeby to zrealizować