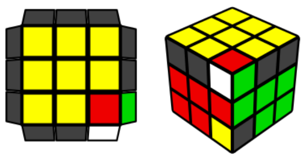
Easy CLS Algorithms
(Corner Last Slot)

Developed by Feliks Zemdegs

Images sourced from Conrad Rider's VisualCube - http://cube.crider.co.uk/visualcube.php

Algorithm Presentation Format



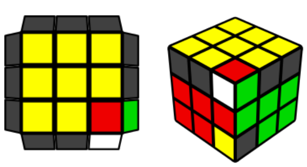
Suggested algorithm here

Round brackets are used to segment algorithms to assist memorisation and group move triggers.

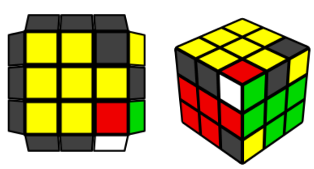
CLS is a large algorithm set. This algorithm sheet presents a subset of CLS algorithms which are either easy to recognise, learn, or execute. This selection of 'easy' cases is based on subjective judgement and experience, and we're always happy to take suggestions about the inclusion/removal of certain cases.

This sheet provides a nice introduction to the full CLS algorithm set, and aims to highlight the most useful cases to know.

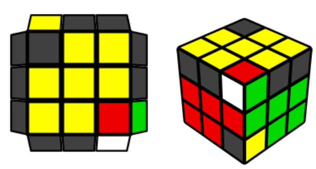
Corner sticker facing front



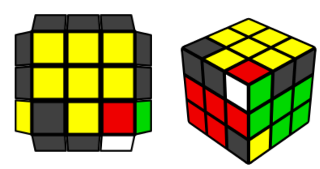
y (R' D R) U' (R' D' R)



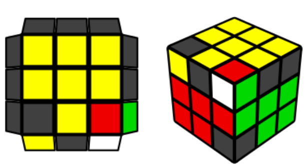
U' (R U' R') (U' R U2 R')



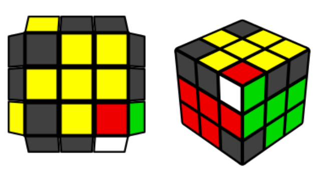
y' U (R' U2 R) (U' R' U' R)



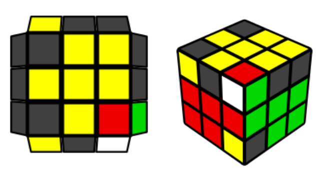
(U L U') R (U L' U') R'



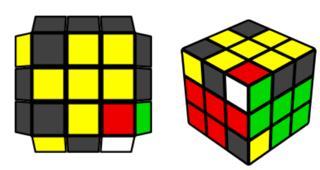
(R U R' U') (R U' R' U)
(R U' R')



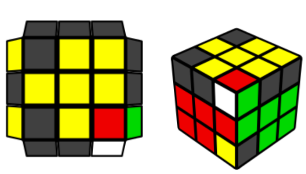
(U' R U' R') U2 (R U R' U')
(R U' R')



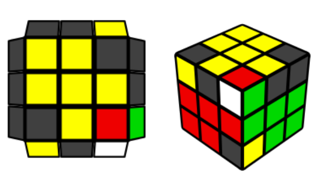
(R U' R) U2' (R' U' R U' R2')



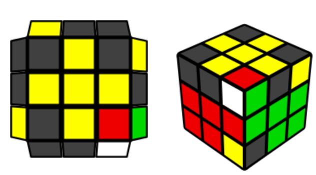
U' (R U' R') (U2 R U' R')



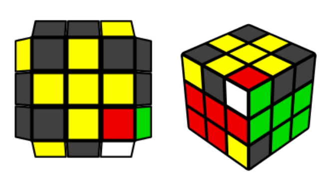
U2 l (R U' R' U) (R U' R' U)
(R U' R' U) l'



y' U' (R' U' R) (U2 R' U' R)

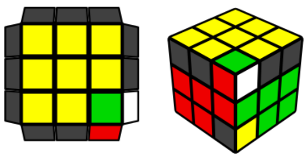


U' (R U' R') (U' R U' R')
(U R U2' R')

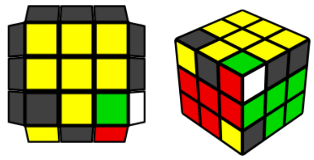


U (R U2' R') (U' R U R')
(U R U R')

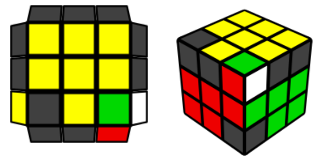
Corner sticker facing right



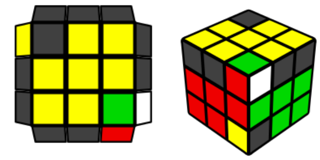
U (R' D' R) (U' R' D R)



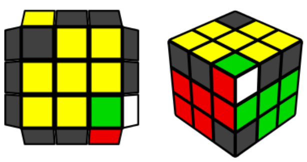
y' U (R' U R) (U R' U2 R)



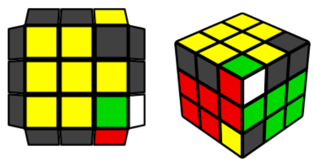
U' (R U2' R' U) (R U2' R' U)
(R U2' R')



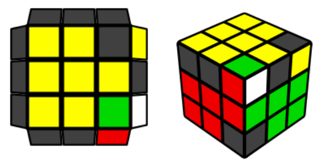
U' (R U2' R' U) (R U R')



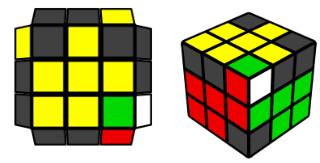
(R U R' U')
(R' F' R U) (R U' R' F)



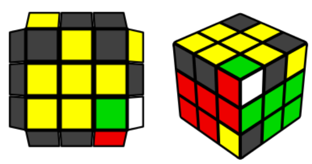
U (R U L) U' (R' U L')



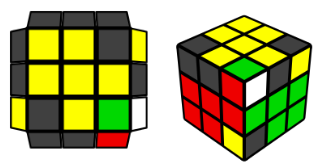
U' (R U R' U') (R U' R' U')
(R U R')



y' U (R' U R U2') (R' U' R U)
(R' U R)

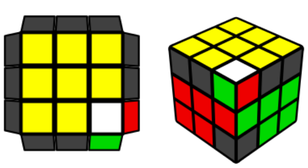


y' U (R' U R U2') (R' U R)

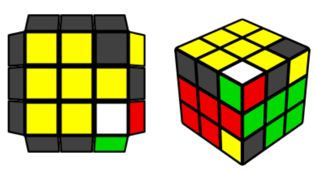


U (R U R') U2' (R U R')

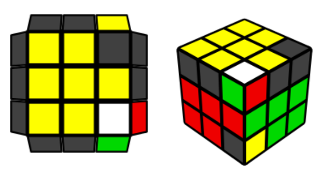
Corner sticker facing up



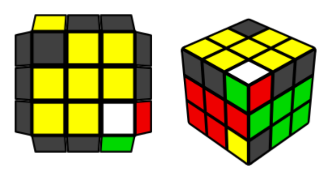
U' (R U' R' U) (R U2' R')
(U' R U R')



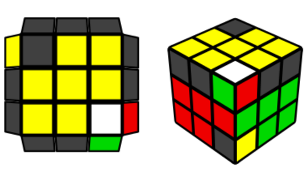
y' U' (R' U2' R U) (R' U2' R U)
(R' U2' R)



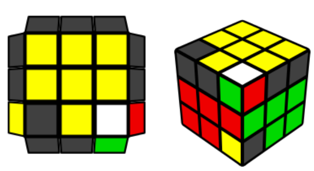
y' U (R' U' R U) (R' U' R U')
(R' U R)



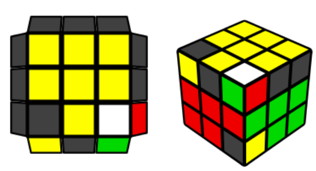
y' U2 (R' U R U') (R' U' R U')
(R' U R)



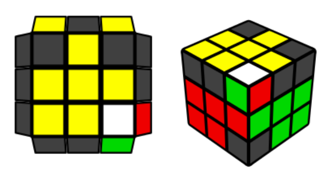
U2 (R U' R' U) (R U R' U)
(R U' R')



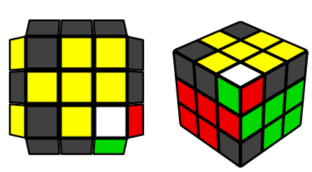
U' (R U R' U') (R U R' U)
(R U' R')



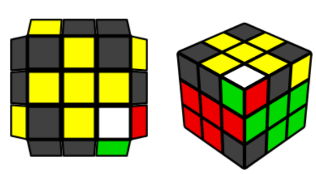
U (R U2 R' U') (R U2 R' U')
(R U2 R')



U (R U' R') (U R U' R')
(U R U' R')

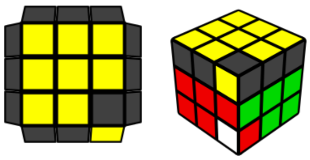


y' U' (R' U R) (U' R' U R)
(U' R' U R)

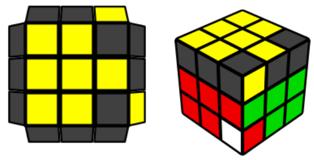


U' (R U R' U') (R U R') U2'
(R U2 R')

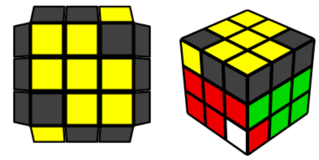
Corner in slot, sticker facing front



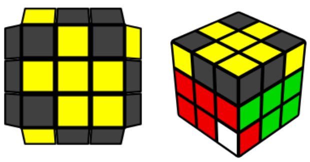
L' (R U R' U') L (R U2 R') (U' R U R')



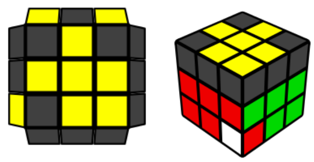
(R U' R' U') (R U R') U2 (R U' R')



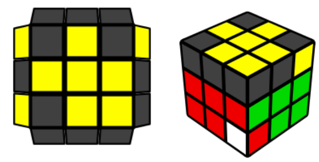
(R U2 R' U2') (R U2 R') (U' R U R')



y' (R' U2 R U') (R' U R) (U' R' U' R)

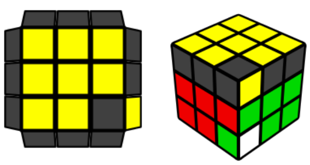


(R U' R' U') (R U R' U') (R U2 R')

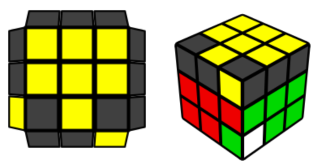


U' (R U R' U') (R U2 R') (U' R U R')

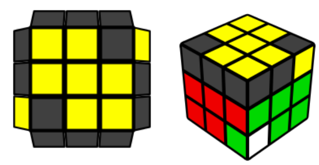
Corner in slot, sticker facing right



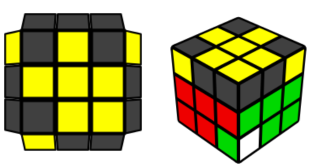
U2 (R U' R' U) (R U2' R') L' (U R U' R') L



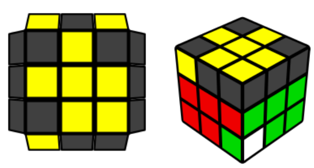
U (R U2' R U) (R' U R) U2' R2'



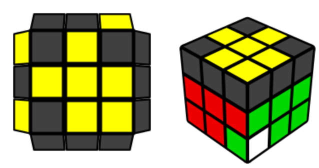
(R U' R' U) (R U2' R') (U2 R U2' R')



(R U2' R' U) (R U' R') (U R U R')



U (R U' R' U) (R U2' R') (U R U' R')



y' (R' U R U) (R' U' R U) (R' U2' R)