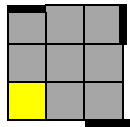


## 2-Look CMLL

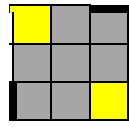
Once the two 1x2x3 blocks are in place, the top corners are solved. CMLL algorithms do this in one step, but here we will orient and position the corners separately.

### Orient corners

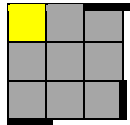
There are seven cases for orienting corners. These corner algorithms are in some cases simpler than those in CFOP because the LL edges and center do not matter.



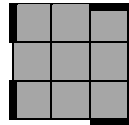
**Sune**  
RUR'URU2R'



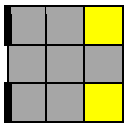
**Bowtie**  
(RU2R'U') (RUR'U')\*2 (RU'R')  
Alt: U2 (FR'F'R) (URU'R')



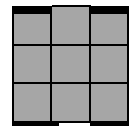
**Anti-Sune**  
R'U'RUR'U2R  
Alt: U2 L'U'LU'L'U2L



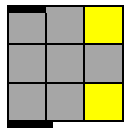
**Pi**  
F (RUR'U')\*2 F'



**Headlights**  
F (RUR'U') F'



**Double Headlights**  
(RU2R'U') (RUR'U') (RU'R')  
Alt: F (RUR'U')\*3 F'



**Blinkers**  
(RUR'U') (R'FRF')

### Position corners

There are two cases to position the top corners. Both can be solved with familiar PLL algorithms.

If there are matching corners (corners with the same color sticker on the same side), then turn U to put the matching corners on the left and use a J Perm:

- (RUR'F')(RUR'U')(R'FR2U') R' (the final U' is unnecessary)

If there are no matching corners, use a Y Perm with any U position:

- F (RU'R'U')(RUR'F')(RUR'U')(R'FRF')