I am not sure that this particular challenge should be listed under "easy" as written. All the other challenges up to this point have been straightforward "in the head" problems. This one requires quite a bit more set-up.

The "easy" portion of this task is to evaluate cost as defined is simply comparing lists of lists (in python, anyway). Generating the matrix of possible magic squares is far more time-consuming than anything up to this point, and might best have been it's own "Generate 3x3 Magic Squares" challenge, with "Magic Square Forming" following directly after.

Were I to see this particular excercise in a "team-coding" evaluation for a job interview, my first suggestion to my partner would be to google for the list of possible 3x3 magic squares and just use that as a look up table, since the boundaries are well understood and fast and lazy will always win!

class Magic(object):

pre = [

[[8, 1, 6], [3, 5, 7], [4, 9, 2]],

[[6, 1, 8], [7, 5, 3], [2, 9, 4]],

[[4, 9, 2], [3, 5, 7], [8, 1, 6]],

[[2, 9, 4], [7, 5, 3], [6, 1, 8]],

[[8, 3, 4], [1, 5, 9], [6, 7, 2]],

[[4, 3, 8], [9, 5, 1], [2, 7, 6]],

[[6, 7, 2], [1, 5, 9], [8, 3, 4]],

[[2, 7, 6], [9, 5, 1], [4, 3, 8]],

]

def evaluate(self, s):

totals = []

for p in self.pre:

total = 0

for p\_row, s\_row in zip(p, s):

for i, j in zip(p\_row, s\_row):

if not i == j:

total += max([i, j]) - min([i, j])

totals.append(total)

return min(totals)