**Peer Review**

Starting with the outcome of your code, both of our solutions seem to be calculating the same results, which is good! It might be a good idea however to print out the points vector (and maybe placements) to easier get an understanding of what the code has calculated. I added a couple of arguments to your output and found it interesting that the solutions were slightly different even though the score was the same. It might have to do with your use of “int search” versus my “solve maximize”, which I also suspect is the reason your code runs slightly faster than mine. A good implementation in other words!

Yet, it could be faster. For the points vector, you use if statements to get the correct values into the vector, but these statements are slow. A faster solution would be to use and-statements, where you simply state (in pseudo code) “first half of the vector (i to n) = rows /\ second half (n+i to n\*2) = columns” (or the other way around).

As for the rest of the code, it is short, concise and easy to read. The use of 0/1 integers instead of true Booleans is good since Booleans slow things down. Lex2 for breaking rotational symmetry is a good choice of constraint, any more might make the code less efficient. All in all the code is very well written, a small tweak might make it marginally faster but overall there’s not much to complain about.