We would make 2 more 2d arrays using the matrices that we have. The first array would contain all the interns in the 0th column and then for each row, the indexes [1-n] would contain the preference number of Employers [1-n] respectively i.e [1,5] would contain the preference number of intern 1 towards employer 5. So from the given example table, this number would be 3, which is the index where E_5 is on I_1's row. We call this array T1.
Similarly a 2d array of employers preference towards all interns would be made in the same manner. In this table, T2, the value of T2[2,6] = 1 because E_2 has I_6 on its 1st index, considering the example table.
To get the answer of a general query 'Does I_x prefer E_y over E_z?' we would compute T1[x,y]-T1[x,z] and if the answer is greater than 0 then return true else return false. This computation takes O(1) time.
For the query 'Does E_x prefer I_y over I_z?' we would compute T2[x,y]-T2[x,z] and if the answer is greater than 0 then return true else return false. This computation also takes O(1) time.
These 2d arrays, being nxn, would both require $O(n^2)$ space each.