1. Good Pairs

[Number of Good Pairs - LeetCode](https://leetcode.com/problems/number-of-good-pairs/description/)

Proposed Solution

int numIdenticalPairs(int\* nums, int numsSize){

    if(numsSize == 1)

        return 0;

    if(numsSize == 2)

    {

        if(nums[0] == nums[1])

            return 1;

        return 0;

    }

    //Initialise the variables

    int i=0,j=i+1;

    int pair\_count = 0;

    while((i < numsSize) && (j < numsSize))

    {

        if(nums[i] == nums[j])

        {

            pair\_count++;

        }

        j++;

        if(j == numsSize)

        {

            i++;

            j = i + 1;

            printf("%d %d\n",i,j);

        }

    }

    return pair\_count;

}

Runtime 7ms

Memory 5.8MB

To be improved:

Usage of loops

Improved solution – Use 2 for loops

int numIdenticalPairs(int\* nums, int numsSize){

int cnt = 0;

for(int i=0;i<numsSize;i++){

for(int j=i+1;j<numsSize;j++){

if(nums[i]==nums[j]){

cnt++;

}

}

}

return cnt;

}