**Merge sort, O(nlongn)**

long long cntSwap;

vector<int> merge( vector<int>& left, vector<int>& right)

{

vector<int> result;

int ll=left.size(), rl=right.size();

int i = 0, j = 0, lmt=ll+rl;

left.push\_back(1e9);

right.push\_back(1e9);

for(int k=0;k<lmt;k++)

{

if(left[i] <=right[j])

{

result.push\_back(left[i]);

i++;

}

else

{

result.push\_back(right[j]);

j++;

cntSwap+=(ll-i);

}

}

return result;

}

vector<int> merge\_sort(vector<int>& vec)

{

if(vec.size() == 1) return vec;

vector<int>::iterator middle = vec.begin() + (vec.size() / 2);

vector<int> left(vec.begin(), middle);

vector<int> right(middle, vec.end());

left = merge\_sort(left);

right = merge\_sort(right);

return merge(left, right);

}