归并排序、求逆序对

每当在左空之前拿右，逆序对+=左余下的数量

存数下标从1开始，需要将第n+1个数置为inf：num[n + 1] = 233333333;

排序完成后逆序对数量存在reverse_num中

```
long long int reverse_num = 0;
int num[MAXN] = { 0 };
int temp[MAXN];
int n;
//归并:
void merge(int l, int r)
{
    int mid = (l + r) / 2;
    int i = l, j = mid, p = l, L_remain = mid - l;
    while (i < mid && j <= r)
    {
        if (num[i] <= num[j])
        {
            temp[p++] = num[i++];
            L_remain--;
        }
        else
        {
            temp[p++] = num[j++];
            reverse_num += L_remain;
        }
    }
    //move remain to temp
    while (i<mid)
        temp[p++] = num[i++];
    while (j<=r)
        temp[p++] = num[j++];
    //move from temp to num
    for (p = l; p <= r; p++)
        num[p] = temp[p];
}

//归并排序：递归左->递归右->归并
//下标从1开始，切分中间点为(左+右)/2,最小基本问题为一个数(左-右)=1
void merge_sort(int l, int r)
{
    //printf("<%d,%d>\n", l, r);
    if (r - l < 1)  return;
    if (r - l == 1)
    {
        if (num[l] > num[r])
        {
            reverse_num++;
```

```
                int temp = num[l]; num[l] = num[r]; num[r] = temp;
            }
            return;
        }
        merge_sort(l, (l + r) / 2);
        merge_sort((l + r) / 2, r);
        merge(l, r);
    }
```