

分治法

思路

- **基本步骤:** 将问题分成若干个(一般都是 2 个 =。=)子问题, 这些子问题的解决互不影响 -> 递归地解决每个子问题 -> 将子问题的解结合得到原问题的解;
- **优势:** 将 $O(n^2)$ 的复杂度降到 $O(n\log n)$.

典型问题

1. 归并排序

问题描述: 就是排序嘛 23333

解决方法: 递归地归并排序左半部分与右半部分, 然后合并它们的排序结果。

2. 逆序对数目

问题描述: 有一排数 $a_1, a_2, a_3, \dots, a_n$, 下标是 $1 \sim n$, 逆序对的定义是 $i < j$ 但 $a_i > a_j$, 求出逆序对的数目。

解决方法: 伪代码如下:

```
# input: List L
# output: Number of inversions in L and sorted list of elements L'

SORT-AND-COUNT (L)

IF list L has one element
    RETURN (0, L).
DIVIDE the list into two halves A and B.
(rA , A) ← SORT-AND-COUNT(A).
(rB , B) ← SORT-AND-COUNT(B).
(rAB , L') ← MERGE-AND-COUNT(A, B).

RETURN (rA + rB + rAB , L').
```

代码如下 (第三次作业第一题):

```

int merge_and_count(vector<int> &arr, int s, int m, int e) {
    vector<int> arr_copy(e - s);
    int i = s, j = m, a_c_i = 0, count = 0;
    while (i < m && j < e) {
        int ai = arr[i], bj = arr[j];
        if (ai > bj) {
            arr_copy[a_c_i++] = bj;
            count += m - i;
            j++;
        } else {
            arr_copy[a_c_i++] = ai;
            i++;
        }
    }
    if (i != m) for (int x = i; x < m; x++) arr_copy[a_c_i++] = arr[x];
    else if (j != e) for (int x = j; x < e; x++) arr_copy[a_c_i++] = arr[x];
    for (int x = s, y = 0; x < e; x++, y++) {
        arr[x] = arr_copy[y];
    }
    return count;
}

int sort_and_count(vector<int> &arr, int s, int e) { // [s, e)
    if (s == e - 1) return 0;
    int m = (s + e) / 2;
    int c1 = sort_and_count(arr, s, m);
    int c2 = sort_and_count(arr, m, e);
    int c3 = merge_and_count(arr, s, m, e);
    return c1 + c2 + c3;
}

```

3. 平面上的最近点对

问题描述: 平面上有许多点点，找出它们之间的最小距离。这里我们假设它们的 x 坐标各不相同。

解决方法: 采用分治法解决的思路如下：

- **Divide:** 将点集根据 x 坐标排序并分成两部分，每个部分点的数目是 $N / 2$ ，中点记为 mid ；
- **Conquer:** 递归地找出两个部分的点对最小距离，并将这 2 个数的最小值记为 min ；
- **Combine:** 选择 x 坐标在区间 $[mid.x - min, mid.x + min]$ 内的点集，根据 y 坐标排序，从最下面的点开始，向上查找 11 个点，比较得到最小距离。

trick: 不要每次都根据 y 方向排序一次，可以在 Divide 时排序一次，在 Combine 的时候直接归并

4. 找第 k 大的数

问题描述: 给一组数，找出其中第 k 大的，希望在 $O(n)$ 的时间内完成。

解决方法: Median-of-medians