

归并排序

归并排序:

59051

1. $[L, R] \Rightarrow [L, \text{mid}], [\text{mid} + 1, R]$
2. 递归排序 $[L, \text{mid}]$ 和 $[\text{mid} + 1, R]$
3. 归并, 将左右两个有序序列合并成一个有序序列

rixClou... 小游戏备选 django 区块链 google 上海落户 Latex公式 iost 推荐系统 直播 精美网站 毕设

```
6
7 const int N = 100010;
8
9 int n;
10 int q[N], tmp[N];
11
12 LL merge_sort(int l, int r)
13 {
14     if (l >= r) return 0;
15
16     int mid = l + r >> 1;
17     LL res = merge_sort(l, mid) + merge_sort(mid + 1, r);
18
19     // 归并的过程
20     int k = 0, i = l, j = mid + 1;
21     while (i <= mid && j <= r)
22         if (q[i] <= q[j]) tmp[k++] = q[i++];
23         else
24         {
25             tmp[k++] = q[j++];
26             res += mid - i + 1;
27         }
28     // 扫尾
29     while (i <= mid) tmp[k++] = q[i++];
30     while (j <= r) tmp[k++] = q[j++];
31
32     // 物归原主
33     for (int i = l, j = 0; i <= r; i++, j++) q[i] = tmp[j];
34
35     return res;
36 }
37
38 int main()
```

I

```
#include <iostream>

using namespace std;

typedef long long LL;

const int N = 1e5 + 10;

int a[N], tmp[N];

LL merge_sort(int q[], int l, int r)
{
    if (l >= r) return 0;

    int mid = l + r >> 1;
```

```

LL res = merge_sort(q, l, mid) + merge_sort(q, mid + 1, r);

int k = 0, i = l, j = mid + 1;
while (i <= mid && j <= r)
    if (q[i] <= q[j]) tmp[k ++ ] = q[i ++ ];
    else
    {
        res += mid - i + 1;
        tmp[k ++ ] = q[j ++ ];
    }
while (i <= mid) tmp[k ++ ] = q[i ++ ];
while (j <= r) tmp[k ++ ] = q[j ++ ];

for (i = l, j = 0; i <= r; i ++, j ++ ) q[i] = tmp[j];

return res;
}

int main()
{
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i ++ ) scanf("%d", &a[i]);

    cout << merge_sort(a, 0, n - 1) << endl;

    return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/39791/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。