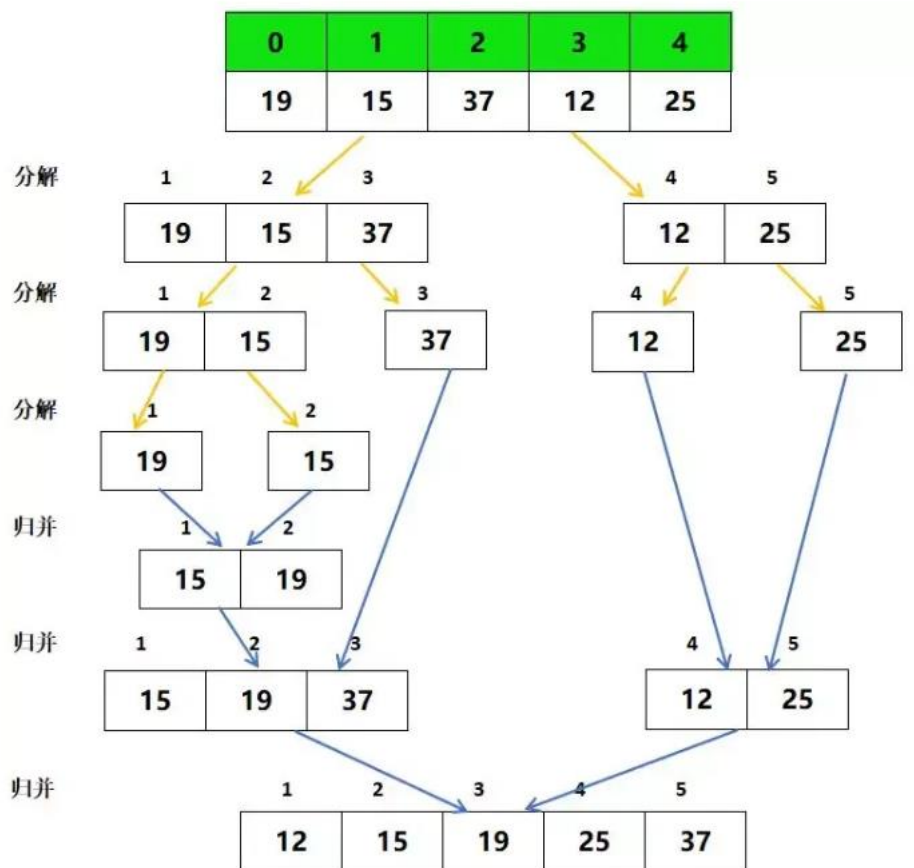


归并排序流程图如下所示（理解大致思想即可）：



源代码如下所示（具体思路和分析均已包含在注释中）：

```
#include <iostream>
using namespace std;

long long ans = 0; // 设置逆序数为全局变量

void Merge(int a[], int left, int right)
{
    if (left == right) {
        return;
    }
    int mid = (left + right) / 2;
    // left 和 mid 分别是要合并的第一个数列的开头和结尾，mid+1 和 right 分别是
    // 第二个数列的开头和结尾
    // i、j 分别标记第 1 和第 2 个数列的当前位置，k 是标记当前要放到整体的哪一个
    // 位置
```

```

int i = left, j = mid + 1, k = left;
int *temp = new int[right - left + 1];
Merge(a, left, mid);
Merge(a, mid + 1, right);
//如果两个数列的数都没放完，循环
while (i <= mid && j <= right)
{
    if (a[i] < a[j]) {
        temp[k++] = a[i++];
    }
    //如果出现逆序数(a[i]>a[j]), 则从 i 到 mid 这 mid-i+1 个数都会比
a[j]大, 均为逆序数, 更新 ans
    else {
        temp[k++] = a[j++];
        ans += mid - i + 1;
    }
}
//放完后, 将剩下没有参与比较的数直接放进 a 数组
while (i <= mid) {
    temp[k++] = a[i++];
}
while (j <= right) {
    temp[k++] = a[j++];
}
for (int i = left; i <= right; ++i) {
    a[i] = temp[i];
}
}

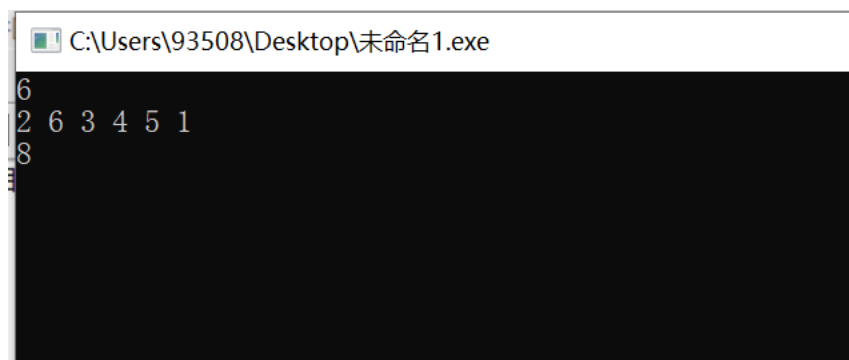
int main()
{
    int n;//代表待排序数字个数
    cin >> n;
    int *a = new int[n + 1];//为方便排序申请 n+1 个空间, 序号 1-n (舍弃 0 下标
这一个空间)
    //输入数组内容
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
    }
    //归并排序
    Merge(a, 1, n);
    //输出逆序数
    cout << ans << endl;
}

```

输出截图如下所示：



```
C:\Users\93508\Desktop\未命名1.exe
5
3 1 4 5 2
4
-----
Process exited after 13.81 seconds with return value 0
请按任意键继续. . .
```



```
C:\Users\93508\Desktop\未命名1.exe
6
2 6 3 4 5 1
8
```