

思路

归并排序是一种高效的排序算法，利用分治的思想，分而治之，也就是将要排列的数分成两半，然后对这两个序列进行排序，然后再合并，当然这是一个递归的过程。其中合并过程稍微复杂一点，举个例子，假设要合并A,B两个序列，从小到大，由于A,B两个序列是排好序的，即都按照从小到大排列，所以每次只要比较A,B中最小的数字，把小的复制到辅助空间，下次比较忽略已经复制过的数字（下标后移），直到合并完成，然后再将辅助空间里的数字复制回原数组空间。代码的话，紫书写的非常简洁，一开始我还误解了它的参数，所以我还是按照自己的想法去写了一个代码（以从大到小排序为例），当然和原书代码思路一致。

代码

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

void merge_sort(int* num, int start, int end, int* temp){
    if(end - start == 0) return;
    //下面是划分
    int mid = (start + end) / 2;
    merge_sort(num, start, mid, temp);
    merge_sort(num, mid+1, end, temp);
    int i = start;
    int j = mid + 1;
    int k = start;
    //下面是合并
    while(i <= mid || j <= end){
        //左边复制结束或者右边最小的小于左边最小的
        if((i > mid) || (j <= end && num[j] < num[i]))
            //复制右边
            temp[k++] = num[j++];
        else
            //复制左边
            temp[k++] = num[i++];
    }
    //复制回原数组
    for(i = start; i <= end; i++)
        num[i] = temp[i];
}

int main(){
    int n;
    cin >> n;
    int num[n], temp[n];
    for(int i = 0; i < n; i++)
        cin >> num[i];
    merge_sort(num, 0, n-1, temp);
    for(int i = 0; i < n; i++)
        cout << num[i] << " ";
    return 0;
}
```