第二章 归并排序

```
#include <iostream>
#define INT_MAX 0x7fffffff
using namespace std;
template<class T>
void Merge(T array[], int start1, int end1, int end2){
  int n1 = end1 - start1 + 1;
  int n2 = end2 - end1;
  T *Left = new T[n1+1];
  T *Right = new T[n2+1];
  int i, j;
  for (i = 0; i < n1; i++){
    Left[i] = array[start1 + i];
  }
  for (j = 0; j < n2; j++){
    Right[j] = array[end1 + 1 + j];
  }
  Left[n1] = INT_MAX;
  Right[n2] = INT_MAX;
  i = 0;
  j = 0;
  for (int k = start1; k <= end2; k++){
    if (Left[i] < Right[j]){
      array[k] = Left[i];
      i++;
    }
    else{
      array[k] = Right[j];
      j++;
    }
  }
  delete Left, Right;
}
template<class T>
void MergeSort(T array[], int start1, int end2){
  if (start1 < end2){
    int end1 = (start1 + end2) / 2;
    MergeSort(array, start1, end1);
```

```
        MergeSort(array, end1 + 1, end2);
        Merge(array, start1, end1, end2);
    }
}
int main(){
    int n;
    cout << "输入数组长度：" << endl;
    cin >> n;
    int *arrays = new int[n];
    cout << "输入数组的各个数字：" << endl;
    int j = 0;
    while (j < n){
        cin >> arrays[j];
        j++;
    }
    MergeSort(arrays, 0,n-1);
    j = 0;
    while (j < n){
        cout << arrays[j] << " ";
        j++;
    }
    cout << endl;
    delete arrays;
    return 0;
}
```