

第四章 最大子数组问题分治法

```
#include <iostream>
#include <limits>
using namespace std;
template<class T>
T FindMaxCrossingSubarray(T array[], int low, int mid, int high, int *leftSubscript, int *rightSubscript){
    T leftSum = INT_MIN;
    T sum = 0;
    int maxLeft;
    int maxRight;
    for (int i = mid; i >= low; i--){
        sum = sum + array[i];
        if (sum > leftSum){
            leftSum = sum;
            maxLeft = i;
        }
    }
    T rightSum = INT_MIN;
    sum = 0;
    for (int j = mid; j < high; j++){
        sum = sum + array[j];
        if (sum > rightSum){
            rightSum = sum;
            maxRight = j;
        }
    }
    *leftSubscript = maxLeft;
    *rightSubscript = maxRight;
    return(leftSum + rightSum);
}

template<class T>
T FindMaximumSubarray(T array[], int low, int high, int *leftSubscript, int *rightSubscript){
    T leftSum;
    T rightSum;
    T crossSum;
    int leftLow, leftHigh, rightLow, rightHigh, crossLow, crossHigh;
    if (low == high){
        *leftSubscript = low;
```

```

    *rightSubscript = high;
    return array[low];
}
else{
    int mid = (low + high) / 2;
    leftSum = FindMaximumSubarray(array, low, mid, &leftLow,&leftHigh);
    rightSum = FindMaximumSubarray(array, mid + 1, high, &rightLow, &rightHigh);
    crossSum = FindMaxCrossingSubarray(array, low, mid, high, &crossLow,&crossHigh);
    if (leftSum >= rightSum&&leftSum > crossSum){
        *leftSubscript = leftLow;
        *rightSubscript = leftHigh;
        return leftSum;
    }
    else if (rightSum >= leftSum&&rightSum >= crossSum){
        *leftSubscript = rightLow;
        *rightSubscript = rightHigh;
        return rightSum;
    }
    else{
        *leftSubscript = crossLow;
        *rightSubscript = crossHigh;
        return crossSum;
    }
}
}

```

```

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++;
    }
    int leftSubscript;
    int rightSubscript;
    int sum = 0;

```

```
sum = FindMaximumSubarray(arrays, 0, n - 1, &leftSubscript, &rightSubscript);  
cout << leftSubscript<<" "<<rightSubscript<<" "<<sum << endl;  
delete arrays;  
return 0;  
}
```