

最大子序列问题

代码部分

- 初步代码

```
int maxSubArray(int* nums, int numsSize){
    if(numsSize == 2)
        return nums[0] + nums[1];
    int sum = nums[0] + nums[1];
    int max = sum;
    int temp = 0;
    for(int i = 2; i < numsSize; i++){
        sum += nums[i];
        temp = nums[i-1] + nums[i];
        if(temp > sum && temp > max_sum)//目前总和与后二者之和比较
            sum = temp;
        if(max_sum < sum)
            max_sum = sum;
    }
    return max_sum;
}
```

问题部分

- 算法

采用贪心算法。如果前面所有元素之和不如最新的两个元素之和，则直接舍弃同时进行更新。

- 时间复杂度

一层循环内不再有任何循环，且易看出核心语句执行次数与numsSize同步，故时间复杂度为 $O(n)$ 。

- 更优算法

无

- 求出k与j

若要求出k与j，则需要实时更新k与j的位置。

```
int maxSubArray(int* nums, int numsSize){
    .....
    int k = 0;
    int j = 1;
    for(.....){
        .....
        if(temp > sum && temp > max_sum){
            sum = temp;
        }
    }
}
```

```
        k = i - 1; //如果更新记录下新的初始位置
    }
    if(max_sum < sum){
        max_sum = sum;
        j = i; //如果更新记录下新的结束位置
    }
}
return max_sum;
}
```