题目

优化问题：

给一组参数，在满足一些列相关条件(约束)下，使设计目标达到最优值，最优化问题通常可以表示为数学规划形式的问题。

正如本题，给的是一个数组，满足条件为子序列和最大。

一般优化问题可以用动态规划来解决。

首先我们要弄清楚的是子问题的格式，当我们试图提出递归关系时，子问题的格式会有很大帮助。

MaxSubArray为当前数列子序列最大的和。

首先想到的子问题格式为maxSubArray(nums,i,j)即从nums[i-j]的maxSubArray，最终目标为maxSubArray(nums,0,nums.length-1),但是这种字问题的格式很难找到与最终问题的关系，也就是说我们不能把原问题化成相应的子问题。

MaxSubArray(nums,i)即从nums[0-i]的maxSubArray，从0开始，到i结束。这样我们就可以把数列从头到尾依次分成n个数列，每个数列在前一个数列的基础上+1个数据，到n个数列的时候正好为原数列。我们从第一个数列开始计算当前数列的最大的和。

我们建立一个用于保存每个数列子序列最大和的数组。这个长度和原序列长度相等。然后开始计算一个子问题。第一个子问题包含第一个数据，自然最大序列和为第一个数据值。

然后我们思考第二个子问题的最大子序列和怎么算呢？首先我们思考如果前一个子问题找到了最大值序列，当前子问题序列只是多了一个数据，就一个数据，我们不用给额外增加的数据再次找个最大子序列和，因为就一个数据，也没法找，我们应该怎么办呢？加上之前子问题的结果啊，我们就合成新的问题的结果了。直接加入吗？不，我们肯定要对情况做出处理。如果上一个子问题的最大子序列和为负数，加入我们当前子序列和只会缩小当前问题的子序列和，这个时候我们不应该把之前子问题的结果加入到当前问题结果中来，如果大于0，就可以直接加入到当前问题中来。我们将每个子问题的结果对应保存在数组中，这样用于后边子问题解决。

我们注意，这道题一定不是将多余的数据加入到之前子问题中，动态规划的思想是在之前问题上处理当前问题，所以我们一定是在之前找到子序列和之上修改当前问题，而不是将当前问题的条件加入到原来问题中。

这是我现在对动态规划的了解，还不是很明白，只是限于这道题。

算法常用的有五大算法，需要花时间去了解一下。有机会会写个总结。

代码

public int maxSubArray(int[] nums){

int n = nums.length;

int[] dp = new int[n];

dp[0]=nums[0];

int max = dp[0];

for(int i =1;i<n;i++){

dp[i] = (dp[i-1]>0?dp[i-1]:0)+nums[i];

max = Math.max(dp[i],max);

}

return max;

}