**实验报告**

**一、802题正确性证明**



**二、思路说明**

首先将区间按照它们的结束时间从早到晚进行排序。

对于某个活动j，定义p[j]表示排在活动j之前的所有活动里面离j最近并且相容的活动，如果没有相容的活动，则p[j] = 0。

定义M[i]，表示从i个活动{1, 2, …, i}中能够得到的不重叠的子集的最大权值和。

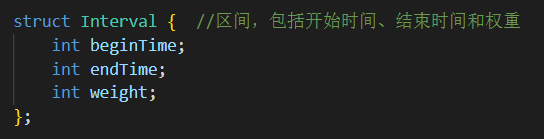
当求解M[n]时，有两种选择：如果不保留区间n，则选择的区间组合没有发生变化，与{1, 2, …, n-1}中能够获取的最大权重和相等，即M[n] = M[n-1]；如果保留区间n，则必须去掉{1, 2, …, n-1}中所有跟n冲突的区间，并且由p数组的定义，即只能从区间{1, 2, …, p(n)}中选取要参加的区间组合，因此M[n] = M[p[n]] + vn

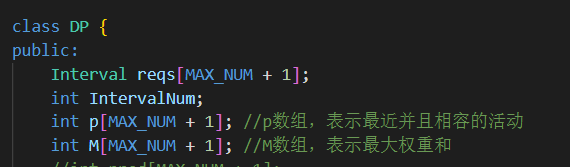
因此，得到M数组的递推公式：M[n] = max( M[p[n]] + vn, M[n-1])。根据该递推公式即可求得最大权值和。

此外，值得注意的是，为了减小时间复杂度，在初始化p数组时我使用了二分查找的方法，使复杂度降低为O(nlogn)。并且在最终求解权值和时将函数的递归转化成了for循环迭代的方式以减少用时。

**三、代码说明**

首先是基本数据结构的定义，包括区间、p数组和M数组等





在DP类内添加成员函数，用于求解本问题。其中prepare函数做预处理，先对所有区间根据结束时间排序，然后初始化p、M数组。binarySearch函数则用二分查找法来确定p数组的值。solveWIS函数则迭代求解M数组的值，最终得到最大权值和。

