

## 1 算法分析

- 第二题是第三题在  $value = 1$  下的特例.
- 第三题是一个典型的动态规划问题, 其子问题是
- 在各个  $end\_time$  之前, 能够取到的最大值
- 在  $end\_time$  取到  $max\_end\_time$  时, 为所求  $max\_value$
- 递推方程为  $DP[end\_time] = \max_{t \leq begin\_time} (DP[t] + value, DP[prev(end)])$
- 递推方程求解  $O(N)$  次, 每一次为二分查表, 耗时  $O(\log N)$ , 共计  $O(N \log N)$

## 2 伪代码

1. if Problem.No is 2, assign 1.0 to all values
2. sort  $node(beg, end, value)$  by  $end$ , increasingly
3. let  $max = 0$
4. for each  $node$
5.   if  $DP[beg] + value > max$
6.     update  $max \leftarrow DP[beg] + value$
7.   insert  $DP[end] \leftarrow max$
8. return  $max$

## 3 代码

见 interval\_cover 文件夹内容, 推荐使用cmake编译

## 4 测试样例

使用Google Test进行单元测试, 请查看test.cpp (包含一个二进制文件, 有大量测试数据)