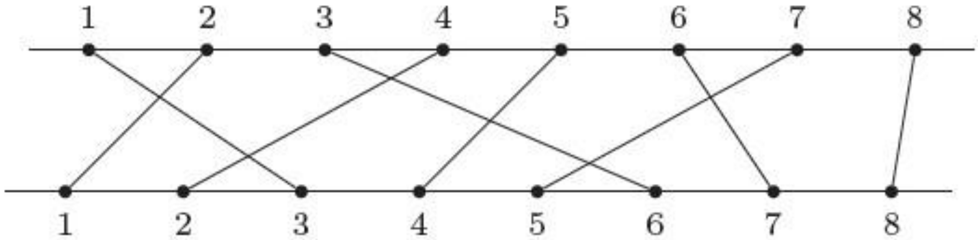
计算机算法设计与分析

朱双贺 2010E8009070012

第六次作业：动态规划解最大相容线段集合问题

题目：在两条互相平行的直线上分别有按顺序排列的 n 个点，标有 1,2，……，n，如下图所示，上面直线上的每一点 i 分别与下面直线上唯一一点π(i)相连，反之亦然，也就是说需要 n 条线段(i,π(i))来连接着 n 对点。



其中，对于任意两条线段(i,π(i))和(j,π(j))，若 i<j 且π(i)> π(j)，或者 i>j 且π(i)< π(j)，则这两条线段必然相交。不满足上述条件，即不相交的线段称为相容线段。试设计一个时间复杂度为 $O(n^2)$的动态规划算法找到这 n 条线段中的最大相容线段集合，即该集合中线段互不相交且线段条数最多。例如上图中 {(1,3),(3,6),(6,7),(8,8)}即为一个相容线段集合（但不一定是最大相容线段 0 集合）。

解：

动态规划：

暂时未想出复杂度为 n^2 的动态规划算法

贪心方法：

1. 求出与每一条线段不相容的线段的条数的最小值，存入数组 N[] 中；
2. 数组 P[]存放最终的结果，即最大相容线段集合；
3. 第一轮，找出 N[]中最小的元素 N[i]，剔除线段 i 以及和线段 i 不相容的线段 a、b、c……，并另 P[1]=i；
4. 第二轮，在余下的线段中找出最小的 N[j]，再剔除线段 j 以及和线段 j 不相容的线段，并令 P[2]=j；
5. 重复上述过程，直到所有的线段都被剔除

算法如下：

Algorithm Find(I[],Π[],n)

```
1.  for i=1;i<=n;i++ do
2.      将I[i]，Π[i]存入一个链表的第i个结点中，该结点的数据域为一个有两个元素
        的数组A[],A[0]= I[i], A[1]=Π[i];结点的指针域指向下一个结点;
3.  end
4.  q=0;
5.  while 链表不为空
6.      找出A[1]最小的结点，q++;
7.      P[q]=该结点的A[0];
8.      找出所有与A[0]线段不相容的线段所在的结点，删除这些节点以及第六步中找出的
        A[1] 最小的结点;
9.  end
```

现在分析算复杂度：

构建链表的复杂度为n；而while循环次数与输入的初始情况有关，最坏情况下，所有线段都相容，则循环将执行n次，每次删除一个节点，这时的复杂度为$n^2/2$；

所以，算法的时间复杂度为：$A(n)\in O(n^2)$