算法基础

第七次作业参考答案（DDL：2024 年 11 月 28 日 23:59）

Q1.（$15+25=35$ 分）

给定三组整数列表 $a_1,\dots,a_n, b_1,\dots,b_n, c_1,\dots,c_n$，判断是否存在 i,j,k 使得 $a_i+b_j+c_k=0$，若存在，请给出一组解。

1. 设计一个最坏时间复杂度尽可能优的算法解决题干中问题。
2. 假设列表中所有元素都在 $[-M,M]$ 的范围内（$M>n$）。设计一个最坏时间复杂度为 $O(M\log M)$ 的算法解决题干中问题。
   提示：第二问可能需要使用快速多项式乘法。

Solution:

1. $O(n^2\lg n)$ 方法：$\forall i,j$，直接计算 a_i+b_j。然后对第三个列表进行排序，会使用 $O(n\lg n)$ 次运算。最终，对在第一步中计算出来的每个和：a_i+b_j，我们进行二分搜索来判断 $-(a_i+b_j)$ 是否在第三个列表中。总的时间复杂度即 $O(n^2\lg n)$。

   $O(n^2)$ 方法：对前两个列表进行排序，然后遍历第三个列表，对每个 c_k，在前两个列表中使用双指针向内搜索法寻找 $a_i+b_j=-c_k$，每次搜索用时 $O(n)$。总用时 $O(n^2)$。

2. 首先，对三个列表进行排序。然后，将所有元素各加 M。此时问题转化为：对于递增非负整数数列 a,b,c，寻找 i,j,k 使得 $a_i+b_j+c_k=3M$。令 S 为 a-列表的元素集，T 为 b-列表的元素集。构造多项式 $A(x)=\sum_{i\in S}x^i, B(x)=\sum_{j\in T}x^j, D(x)=A(x)B(x)$，则在 $D(x)$ 中，x^p 前的系数为：$|\{i\in S, j\in T: i+j=p\}|$。使用 FFT 算法，计算 $D(x)$ 各项系数共用时 $O(M\lg M)$。遍历 $D(x)$ 中各非零项 x^p，在 c-列表中二分搜索寻找 $c_k=3M-p$。一旦找到了某个满足条件的 c_k，则遍历 a-列表各元素 a_i，在 b-列表中二分搜索寻找 $b_j=3M-c_k-a_i$，最终找到的 (i,j,k) 即为一组解。最坏时间复杂度 $O(M\lg M)$。

Q2.（15 分）

当 n 个字符组成的字符集对应的出现次数恰为前 n 个斐波那契数（第 $i(0\le i\le n-1)$ 个字符的出现次数为 F_i）时，求最优前缀编码。

注：$F_0=F_1=1$,$F_j=F_{j-1}+F_{j-2}(j\ge 2)$

Solution:

编码为 $code(c_{n-1}) = 0$，$code(c_{i-1}) = 1code(c_i)$ $(1 \le i \le n-2)$，$code(c_0) = 1^{n-1}$。

证明：

引理 1 $\forall k \in \mathcal{N}$，$\sum_{i=0}^{k} F_i = F_{k+2} - 1$。

引理 2 z 为由哈夫曼算法构建的树 T 的内部节点，则 $z.freq$ 等于以 z 为根的子树上所有叶子结点的频率之和。

递归定义子树 T_n：

1. $T_1.left = c_1, T_1.right = c_0$，$T_1.freq = c_0.freq + c_1.freq = 2$；
2. $\forall i,\ 2 \le i \le n-1, T_i.left = c_i, T_i.right = c_{i-1}, T_i.freq = c_i.freq + T_{i-1}.freq$

由以上性质，可保证树的结构唯一，对应的编码也确定。

Q3. $(10 + 5 + 20 = 35$ 分)

假设在科大超算中心有一系列计算任务 $S = \{s_1, s_2, \ldots, s_n\}$，每个任务 s_i 分别需要执行 d_i 分钟。同时，这些计算任务之间存在着无环的依赖关系 $P = \{(s_i, s_j) | s_i, s_j \in S\}$，其中每条依赖关系 (s_i, s_j) 表示任务 s_j 需要在任务 s_i 完成之后才能开始执行。请为超算中心设计一套计算任务调度算法，输入为 S，$\{d_i\}$，和 P，输出为 $\{x_i\}$，其中 x_i 表示每个任务的开始时间。算法的设计目标是最小化系统运行时间跨度（即从第一个任务开始至最后一个任务结束的时间跨度）。

1. 该问题是一个典型的规划问题，请用数学语言表示该线形规划问题（包括优化目标以及约束条件）。
2. 简要分析利用单纯形法解决前一问所给出的线形规划的时间复杂度。
3. 请给出一个线性时间复杂度的贪心算法，且保证其输出的结果为最优解，并证明其时间复杂度以及最优性。

Solution:

1. 引入虚拟任务 s_0，s_{n+1}，其执行时间均为 0。

   数学模型：

$$\min x_{n+1} - x_0$$

s.t.

$$x_i \geq x_0,\ \forall i \in [1, n]$$

$$x_i + d_i \leq x_{n+1},\ \forall i \in [1, n]$$

$$x_i + d_i \leq x_j,\ \forall (s_i, s_j) \in P$$

2. 最坏时间复杂度 $O(2^n)$

3. 根据依赖关系 P，构建有向无环图。按照拓扑序对所有任务进行遍历，任务在前序依赖任务都完成后立即执行。

   贪心算法的最优性来源于拓扑排序的性质。拓扑排序保证了任务按照依赖关系的顺序执行，且对于每个任务 s_j，其开始时间是其所有前驱任务结束时间的最大值，这样的安排可以确保任务的开始时间尽可能早，同时不会违反依赖关系。

   通过拓扑排序，任务调度的开始时间是按顺序安排的，每个任务的开始时间总是取决于其前驱任务的结束时间，因此没有任何任务被延误到不必要的时间，达到了最优的时间跨度。

Q4.（15 分）

在课程中我们讨论了用于计算有向无环图（DAG）的拓扑排序。在输入图确实是 DAG 的前提下，这个过程将最终生成一个拓扑排序。但是，假设我们给定的图是一个任意的有向图。请扩展拓扑排序算法，使得在给定一个输入有向图 G 时，它输出以下两种情况之一：（a）一个拓扑排序，从而确定 G 是一个 DAG；或（b）G 中的一个环，从而确定 G 不是一个 DAG。你的算法的运行时间应为 $O(m+n)$，其中 n 表示图中的节点数，m 表示图中的边数。

Solution:

执行 topological-sort，若在深度优先搜索过程中找到一条后向边 (u, v)，则在深度优先森林中节点 v 是节点 u 的祖先。因此，图 G 包含一条从 v 到 u 的路径，该路径与后向边 (u, v) 构成了环路。