北京大学 2014 年研究生算法课第 2 次作业

发布时间：2014 年 10 月 13 日
截止时间：2014 年 10 月 27 日课前

注意事项：

- 作业应独立完成，严禁抄袭。作业必须使用统一规定的模板。
- 在截止日期那天，直接把纸质版的作业交给任课老师。
- 如果因生病等特殊原因不能按时完成作业的，那么应在截止日期前一天向任课老师请假。

贪心法

1. **题目来源：**《算法设计》第四章第 16 题：

题目描述：

某工作在金融领域的安全顾问正在调查一种可能的洗钱模式。调查至今已经指出在目前这些天里发生了 n 项可疑的转账，每项都与钱被转入一个单一账户相关。不幸的是，对日期证据的粗略分类意味着他们不知道这个账号，转账的数量，或者转账发生的准确时间。他们仅仅是对每笔转账的近似的时间标签；这个时间标签具有“误差”e_i，转账 i 发生在时间 t_i，满足 $t_i - e_i <= t_i <= t_i + e_i$，注意不同的转账可能有着不同的误差。

在最后一天的前后，他们偶然看到一本银行的账，他们怀疑可能这本帐与这个罪行有关。涉及这本帐有 n 个最近事件，他们发生在时间 $x_1, x_2, x_3, \ldots, x_n$．为了看是否可能这真的就是他们正在找的帐，他们想知道能否把帐上 n 个事件中的每个事件与 n 项可疑转账中的一项不同的转账以这种方式联系起来，如果帐上在时间为 x_i 的事件与近似发生在时间 t_i 的可疑转账联系起来，那么 $|t_i - x_i| < e_j$ (换句话说，他们想知道是否在账上排列的活动伴随着在误差内的可疑转账；这里难处理的部分是他们不知道哪个账上的事件与哪个可疑转账相联系)。

给出一个有效的算法，接收给定的数据并且判定是否存在这样的一种关系。如果可能，你应该使得运行时间至多是 $O(n^2)$.

2. **题目来源：**《算法设计》第四章第 19 题：

题目描述：

一个在通信公司 CluNet 的网络设计组发现他们面对下面的问题：他们有一个连通图 G = (V, E), 其中结点代表想通信的站点。每条边 e 是一条通信链路，具有给定的有效带宽 b_e.

对每个结点 u，v ∈ V，他们想选择一条 u-v 路径 P 以使得这对结点将在上面通信。这条路径 P 的瓶颈值 b(P)是它包含的任何的最小带宽；即 $b(P)=\min_{e\in P} b_e$. 对 G 中一对 u，v 最佳可达瓶颈值就是在 G 中所有 u-v 路径 P 上的 b(P)的最大值。

对每对结点保持路径追踪正变得非常的复杂，于是一个网络设计者提出一个大胆的建议：也许可以找到 G 的一棵生成树 T，使得对每对结点 u，v，在树中的唯一路径 u-v 实际上达到对于 G 中 u，v 的最佳可达瓶颈值（换句话说，即使你能够在整个图中选择任何一条 u-v 路径，你也不能比在 T 中的 u-v 路径做的更好）。

一些天以来这种想法在 CluNet 的办公室中受到全面质疑，并且对于怀疑论存在一个很自然的理由：每一对结点可能想要一个看起来非常不一样的路径来极大化它的瓶颈值；为什么应该只存在同时使得每个人满意的一棵树？但是在排除这个想法的几个企图失败之后，人们开始猜想它也许是可能的。

证明这样的一棵树存在，并且给出一个有效的算法找到它。即给出一个构造一棵生成树 T 的算法，其中对每对结点 u，v，T 中的 u-v 路径的瓶颈值等于对 G 中这对 u，v 的最佳可达瓶颈值。

3. **题目来源：**《算法设计》第四章第 18 题：

题目描述：

你的朋友正在计划下一个寒假深入到加拿大北部的小镇探险。他们已经研究了所有的旅游选择并且画了一个有向图，其结点表示中间目的地且边表示它们之间的路。

在这样一个进程中，他们也熟悉，在世界的这个地区，极端的气候会导致行程变得相当慢并可能引起旅行被大大延迟。他们找到了一个优秀的旅游网站，它可以精确地预报他们沿着这条路将能够旅行得多快；但是旅行的速度在一年中的不同时间是不同的。更准确地说，网站回答下述形式的询问：给定一条连通两个站点 v 与 w 的边 e=（v,w），给定一个从位置 v 的预定开始时间 t，这个网站将返回一个值 $f_e(t)$预测到达 w 的时间。网站保证对所有的边 e 和所有的时间 t，$f_e(t) >= t$ (你不可能在时间上向后旅行)，且 $f_e(t)$是 t 的单调增函数（即你不可能开始得迟反而到达得更早）。除此之外，函数 $f_e(t)$可能是任意的。例如，在旅行时间不随季节改变的区域，我们有 $f_e(t) = t + l_e$, 其中 l_e 是从边 e 的开始到终点的旅行所需要的时间。

你的朋友想利用这个网站通过这个有向图确定从他们的始点到预定的目的地旅行的最快的路（你应该假设他们在时刻 0 开始，并且这个网站做的所有预测都是完全正确的），给出一个求解这个问题的多项式时间的算法，其中我们把对这个网站的一次询问（基于一条特定的边 e 与一个时间 t）看做一步计算。

4. **题目来源：**《算法设计》第四章第 14 题：

注意：这道题的第二问的中文版描述可能比较难读懂。第二问其实是对第一问解题思路的一种提示。如果不明白第二问的含义，那么只要把第一问做对就可以了。

题目描述：

你正在和一组安全顾问一起工作，他们正在帮助监测一个大的计算机系统。对于追踪那些标记为“敏感的”进程有特别的兴趣。每个这样的进程有一个指定的开始时间与结束时间，并且它在这些时间之间连续运行；顾问有一张表，上面有那天将要运行的所有敏感进程的计划开始与结束时间。

作为简单的第一步，他们已经写了一个称作 status_check 的程序，当被调用时就运行几秒钟并且报告关于当时系统上正在运行的所有敏感进程的日志信息（我们将把每个 status_check 的调度建模为当时只在这一个时间点运行，即瞬间内完成的）。他们想要做的是在这一天以最少的次数运作 status_check,但是足以使得每个敏感进程 P，status_check 在进程 P 执行期间至少被调用一次。

(a) 给定一个有效的算法，对于给定的所有敏感进程的开始与结束时间，找一个尽可能小的调用 status_check 的时间集合，使得它满足 status_check 在每个敏感进程 P 执行期间至少调用一次的需求。

(b) 当你设计算法时，安全顾问正在做一点粗略的推理估计。“假设我们可以找到 k 个敏感进程的集合，且具有任何时间都没有两个进程同时运行的性质。那么很清楚，你的算法将需要调用 status_check 至少 k 次：没有一次 status_check 调用可以处理其中一个以上的进程。”

当然这是正确的，在做进一步的讨论之后，你们大家开始想知道是否某些比上述论点更强的结果也是真的。假设 k^*是满足下述性质的最大 k 值：人们可以找到 k 个敏感进程的集合，使得没有两个进程同时运行。那么下述情况成立吗？即一定存在 k^*个时间的集合，在这些时间你可以运行 status_check 以使得每个敏感进程的执行期间都会出现某个调用（换句话说，在前面一段里的这类论点确实是迫使你需要许多次

status_check 调用的唯一理由）. 你认为这个断言是真的还是假的，做出决定，并且给出一个证明或者一个反例。

5. **题目来源：**《算法导论》

题目描述：贪心法反例：

1. 贪心法解决矩阵乘法问题。《算法导论》15.3-5

   有人提出以下贪心算法：在矩阵链 $A_iA_{i+1}...A_j$ 中选择划分点 A_k 使得最后两个矩阵的乘法代价 $p_{i-1}p_kp_j$ 最小。这种贪心法是否能够对所有的实例得到最优解？举出反例或证明你的结果。

2. 关于上题中的矩阵乘法问题，有人提出另一种贪心算法：每次在矩阵链中选取计算代价最小的两个相邻矩阵做乘法，使得矩阵链的长度减一。这样由下而上地逐步计算直至得到最终结果。这种贪心法是否能够对所有的实例得到最优解？举出反例或证明你的结果。

3. 有 n 个分别排好序的整数数组 A_0 ， A_1 ， … ， A_{n-1}，其中 A_i 含有 X_i 个整数，i = 0,1,…,n-1. 已知这些数组顺序存放在一个圆环上，现在要将这些数组合并成一个排好序的大数组，且每次只能把两个在圆环上处于相邻位置的数组合并。定义合并的代价为两个数组中的元素个数之和。问如何选择这 n-1 次合并次序以使得合并时总的代价达到最少？有人如下设计贪心法：计算所有相邻两个数组的元素数之和，从中选择元素之和最小的两个数组进行合并。这种贪心法是否能够对所有的实例得到最优解？举出反例或证明你的结果。