算法课第2次作业

	题目1	题目2	题目3	题目4	题目5	总分
分数						
阅卷人						

1 题目1 洗钱模式

算法 每个洗钱转账都有一个可行区间$[t_i - e_i, t_i + e_i]$,我们把这n个洗钱区间的上界从小到大排序（假设排序后的顺序为$t_1, t_2, ..., t_n$），将n个可疑事件从小到大排序（假设排序后的顺序为$x_1, x_2, ..., x_n$）。则对n个洗钱区间，在$\{x_i\}$里面顺序查找在区间$[t_i - e_i, t_i + e_i]$的第一个候选事件作为t_i的对应事件，循环直到(1)n个t_i都找到配对或者(2)有一个t_i没有找到配对，说明不存在满足条件的配对。

复杂度分析 可行区间排序复杂度$O(n \log n)$，可疑事件排序复杂度$O(n \log n)$；对于每个洗钱区间$[t_i - e_i, t_i + e_i]$，查找对应的可疑事件，复杂度是$O(n)$，所以总的匹配过程的复杂度是$O(n^2)$。总体复杂度为$O(n \log n + n \log n + n^2) = O(n^2)$

正确性证明 下面证明，只要存在可行解，都等价于一个按照上述算法得到的配对。假设有一个配对$P : \{(t_1, x_1), (t_2, x_2), ..., (t_n, x_n)\}$, 设前$k-1$个配对和我们安装上述算法得到的匹配相同，第$k$个不同

$$\{(x_1, t_1), (x_2, t_2), ..., (x_{k-1}, t_{k-1})\}$$

假设在我们的算法得到的配对里t_k和x_j配对。因为有t_k可以和x_j配对，所以t_k的配对是存在的，所以算法进行到第k步也能找到一个解。因为算法是顺序查找到x_j的，说明x_j 是当时集合里最早的可以匹配t_k 的点，所以

$$x_j <= x_k \tag{1}$$

在P中x_j和t_j结合，按照算法的步骤t_j在t_k之后，说明

$$t_j + e_j > t_k + e_k \tag{2}$$

又因为

$$t_j - e_j <= x_j \tag{3}$$

$$t_k + e_k >= x_k \tag{4}$$

由(1)(2)(3)(4)得

$$t_j - e_j <= x_j <= x_k <= t_k + e_k <= t_j + e_j$$

所以我们将R中$(t_k, x_k), (t_j, x_j)$的匹配交换成$(t_k, x_j), (t_j, x_k)$依然是一个匹配，但是前k个都相同。递推可以将前$k+1, k+2, ..., n$个都变为算法得到的满足要求的匹配。所以如果存在解，算法一定得到解。如果不存在解，显然算法得不到解。所以算法是正确的

2 题目2 最优生成树

求最优生成树方法 设有N个点。任取一点作为生成树的起始点n_1,一个已经在生成树的集合A，一个剩余点集R。初始时$R=\{n_1\}$,剩余点集$A=\{n_2,...,n_N\}$。接下来的每个循环中，从A中取一个点n_k加入R中，并在n_k与R中点的所有连线中选取一条加入A，形成一个大小为k的生成树。选线的方法为：初始化为第一条线，形成一棵大小为k的生成树，对于其他n_k与前一循环得到的R中点的连线，将其加入R，必然生成一个环，然后去掉环中权重最小的边。

证明正确性 为了证明正确性，我们根据$u-v$的瓶颈链路的长度采用数学归纳法证明。
不妨假设在构造生成树的时候先加入u点，再加入v点。这里我们假设算法构造的生成树为T_my。
(1)当$u-v$的最优链路长度为1时，证明其在我们算法构造的路径上。反证，假设不在T_my上，则边(u,v)不在T_my上。再加入v点到T_my的时候(u,v) 也在候选线段里，说明在加入时其生成了环且是换上权值最小的边。所以将(u,v)加入T_my后会形成环，这样就有两条$u-v$的不重叠的路径，显然非(u,v)边的那一条路径的每一段的阈值都大于等于边(u,v)，显然非(u,v)边的那一条路径是$u-v$的瓶颈路径，矛盾！所以最优瓶颈路径长度为1时成立
(2)设当$u-v$的最优链路长度小于等于k时成立，证明当$u-v$的最优链路长度为$k+1$时成立。反证。如果$u-v$的瓶颈路径实际为$L_1: u-...-v_11-v_1-v$, 在T_my 得到的瓶颈路径为$L_2: u-...-v_01_v_0-v$。显然v_0在v之前加入T_my（否则v_0加入时不可能同时加入两条边$(v_0,v),(v_01,v_0)$），并且两条路径可以连成一个环C。再根据v_1 和v 加入T_my的顺序讨论：
(a)如果v_1先进T_my，则加入v时$(v,v_1),(v,v_0)$都是候选线段，按照反证假设，肯定会选(v,v_1)，所以T_my 中的瓶颈路径就是$u-...-v_11-v_1-v$，一致
(b)如果v_1后进T_my，则算法在加入v_1时，候选线段有(v_1,v) 但是却没选，说明(v_1,v)加入时是环上值最小的线段，显然环C上所以其他边的值都不比(v_1,v)小，也就是L_2 的每条边权重都不比(v_1,v)小，也就是L_2的阈值肯定不必L_1小，矛盾！所以当最优链路长度为$k+1$时成立(3)综合上面得证

3 题目3 最优旅游线路

求最优旅游线路 对dijkstra算法进行修改后求解。设出发点为u，目的地为v。集合A表示已经计算过的从u出发所需时间的城市列表，$U-A$表示剩余没有找到的城市列表。初始时$A=\{u\}$。每次循环向A中加入一个新点，且新点的旅游时间最短。新点u_k的旅游时间的计算方式为：

$$cost(u_k)=\min_{a_i\in A and (a_i,u_k)}(cost(a_i)+f_e(cost(a_i))$$

其中(a_i,u_k)表示a_i和u_k连边。

证明正确性 对于算法生成的所有点，按照离u点的间隔点数用数学归纳法证明。
(1)当距离u点距离为0时显然成立，就为点u (2)设当间隔点数小于等于k时成立，当间隔点为$k+1$时。反证，假设不成立，设不成立的路径为$L_1: u-u1-...-u_k-u_{k+1}$，则显然$L_1$的从$u$开始的所有子路径都是最优的，由假设知其都是算法生成的生成数上的点。则当算法加入u_{k+1}点进入集合A时没有选择(u_k,u_{k+1})，说明存在一条路径$L_2: u-v1-...-v_t-u_{k+1}$代价更小（$L_1$也在候选路径里），则$L_1$不是最优路径，矛盾。

4 题目4 status_check

(a)**求最小$status_check$集合** 将所有进程按结束时间从早到晚排序，然后依次扫描日志：找到第一个没有被$status_check$集合元素覆盖的进程th_i，取其结束时间end_i作为新的日志检测点插入$status_check$集合，并将th_i之后所有end_i日志检测点可以检测到的进程删去（因为已经被end_i覆盖了），继续下去直到所有进程都被覆盖，则最终生成的$status_check$集合就是最小$status_check$集合

证明上述算法的正确性 我们证明：对于任意一个最优解，都可以转化成按照上述算法得到的解，不改变集合的元素个数。由算法求解结果的唯一性说明算法求解的结果就是一个最优解。
设(a)得到的集合为$S = s_1, ..., s_m$,设一个最优解为$S_0 = s_{01}, .., s_{0n}$。不放设S_0和S的前k个数相同(前k个$status_check$时间点相同)。则对于第$k+1$个时间点$s_{k+1}, s_{0(k+1)}$，并且有s_{k+1}是按照算法的进程th_kj的结束时间。因为S_0前k个检测时间没有覆盖th_kj，说明S_0 中也应该有一个时间点覆盖了th_kj，不妨设为$s_{0(j+1)}$，又s_{k+1}是按照算法的进程th_kj的结束时间，所以$s_{k+1} > s_{0(j+1)}$，且除了前k个时间点覆盖的集合，s_{k+1}覆盖了所有$s_{0(j+1)}$覆盖的集合。将S_0中的$s_{0(j+1)}$替换为s_{k+1}，记新的集合为S_0'。则S_0'等同于S_0的覆盖性，且S_0' 的元素个数也为n，但是S_0'与S的相同元素个数变为$k+1$。重复上述过程可以将S_0转化为S，说明$m = n$，即算法得到的覆盖集合是最小的。

(b)**正确** 设(a)得到的集合为$S = s_1, ..., s_m$，由选取方式可知，这m个检测时间点分别对应一个进程的结束时间，且这m个进程互不重叠，所以最大的互补重叠的进程个数一定大于等于m。而这m个时间点可以覆盖所有的进程，说明任意$m+1$个进程都可以被这m个时间点覆盖，说明至少有一个时间点覆盖了2个进程，即这两个进程相互重叠，所以最大的互补重叠的进程个数一定小于(m+1)。综上，最大的互补重叠的进程个数一定等于最小$status_check$集合的元素个数

5 题目5 贪心法反例

(a)**不能** 设矩阵大小为$A_1 : 1 \times 2$ $A_2 : 2 \times 3$ $A_3 : 3 \times 2$两种选择：$(A_1A_2)A_3$和$A_1(A_2A_3)$，代价分别为$1 \times 2 \times 3 + 1 \times 3 \times 2 = 12$ 和$2 \times 3 \times 2 + 1 \times 2 \times 2 = 16$按照最后一步代价显然选择后者，但是前者的总代价最少，矛盾！

(b)**不能** 设矩阵大小为$A_1 : 2 \times 1$ $A_2 : 1 \times 2$ $A_3 : 2 \times 3$两种选择：$(A_1A_2)A_3$和$A_1(A_2A_3)$，代价分别为$2 \times 1 \times 2 + 2 \times 2 \times 3 = 16$ 和$1 \times 2 \times 3 + 2 \times 1 \times 3 = 12$按照第一步代价显然选择前者，但是后者的总代价最少，矛盾！

(c)**不能** 设数组的元素个数依次为$3, 3, 4, 20, 4$。按照贪心法答案为$(3+3)+(3+3+4)+(3+3+4+4)+(3+3+4+4+20) = 64$按照两个$3-4$先配对的结果是$(3+4)+(3+4)+(3+4+3+4)+(3+4+3+4+20) = 62$，所以贪心法不正确。