HW9

王嵘晟　PB1711614

1.

a.

充分性：因为 $\forall v \in V$，有 $in - degree(v) = out - degree(v)$。所以对于每个结点，出边数等于入边数，则图 G 中每个结点都是偶次的。所以 G 是欧拉图，则强连通有向图 G=(V,E) 中一定有一条欧拉回路。

必要性：因为强连通有向图 G 中有欧拉回路，所以 G 是欧拉图。则 G 没有奇次项。假设 $\exists v \in V$，$in - degree(v) \neq out - degree(v)$ 则一定 $\exists e \in E$，经过 e 后进入 v 没有出边或者经过 e 后离开 v 到了下一个顶点不能继续到再下一个顶点，这与图 G 示欧拉图矛盾。所以 $in - degree(v) = out - degree(v)$。

b.

用 FE 算法遍历图的每一条边来找到欧拉回路，详见算法 1

在 FE 算法中，由于遍历了每一条边，所以算法的时间复杂度为 $O(E)$。

2.

a.

将这个问题转化为图问题，每种货币 c_i 构成图的顶集，任意两种货币间的汇率构成图的边集 (c_i, c_j) 与 (c_j, c_i)，对每条边 $R[i_1, i_2]$ 取对数并取相反数得 $-lnR[i_1, i_2$，以此为边权，将原问题转化为在图中找负权回路问题。使用 Bellman Ford 算法。算法的运行时间为 $O(VE)$，则带入 $|V| = k$, 时间为 $O(\frac{k^2(k-1)}{2})$

Algorithm 1 *

找图 G 的欧拉回路

function Find-Euler-Circuit(G)
    for $(v_i, v_j) \in E$ **do**
        $Make\ (v_i, v_j) white$
    end for
    $Choose\ v_0\ to\ be\ the\ source\ vertex,\ insert\ v\ to\ list\ L$
    while $edge(v_0, v_1)\ comes\ out\ from\ v_0\ is\ white$ **do**
        $Make\ (v_0, v_1)\ black$
        $u = v$
        $Insert\ v_1\ to\ L$
    end while
end function

b.

最初的操作跟 a 过程相同，将原问题转化为在图上找负权回路的问题。然后对每个边做松弛操作，共 $|V|-1$ 次。记录每个顶点的 d 值，然后松弛每个边 $|V|$ 次。然后来检查记录的顶点的 d 值哪些有所减少，对于 d 减少的顶点，都位于可能不相交的负权回路上。用集合 S 来记录这些顶点，为了找到一条回路，对于 S 中的任意一个顶点，用贪心算法来找出它通过边可达的其他所有顶点，用这种方法可以找到一条负权回路。找到这样一个序列后执行打印操作。由于之前的松弛操作的时间为 $O(V)$，执行 Bellman Ford 算法时间为 $O(VE)$，带入得总的时间复杂度为 $O(\frac{k^2(k-1)}{2})$

3.

由引理 25.1，$\hat{\omega}(u,v) = \omega(u,v) + h(u) - h(v)$，所以对于环路 c 来说，重新加权后任意一条环的总权重不变。所以由 $\hat{\omega}(c) = \omega(c) = 0$，重新加权后的图中，环还是之前的环，总权重为 0。由于重新加权后不再存在负权边，可环路的总权重为 0，所以每条边的权重 $\hat{\omega}(u,v) = 0$。

4.

a.

如果对于边 (u,v) 不存在最小切割，则最大流没有办法增加，则剩余的网络中不存在增广路径。所以当增广路径存在时，执行一次 Ford-Fullkerson 迭代，找到并增加该路径，由于边的容量是整数，所以每次增加都是整数。由于每条边的流严格增加，并且每次增加一个整数。所以 Ford-Fullkerson 第三行的 while 循环每次迭代使得流量 +1，直到达到最大流。所以为了找到增广路径，使用 BFS，用的时间为 $O(V+E)$。

b.

如果边的流量比容量小至少 1，则不会有任何变化。否则使用 BFS 在 $O(V+E)$ 的时间内查找从 s 到 t 包含边 (u,v) 的路径，将该路径上每个边的流减小 1，然后在 $O(V+E)$ 的时间内运行 Ford-Fullkerson 算法循环的迭代，由于每次增加的都是整数值，所以最终结果要么是找不到增广路径，要么是最后总的最大流减少 1 然后结束。