```
# Algoritmo de refinamento de cores
# Desenvolvido por Igor Domingos da Silva Mozetic
# RA: 11202320802

import time

def ler_instancias(arquivo):
    with open(arquivo, 'r') as f:
        linhas = [l.strip() for l in f if l.strip()]

    instancias = []
    i = 0
    while i < len(linhas):
        n = int(linhas[i])
        i += 1
        g1 = [[int(x) for x in linhas[i + j]] for j in range(n)]
        i += n
        g2 = [[int(x) for x in linhas[i + j]] for j in range(n)]
        i += n
        instancias.append((n, g1, g2))
    return instancias

def incidencia_para_lista(grafo):
    n, m = len(grafo), len(grafo[0])
    adj = [[] for _ in range(n)]
    for j in range(m):
        u_v = [i for i in range(n) if grafo[i][j] == 1]
        if len(u_v) == 2:
            u, v = u_v
            adj[u].append(v)
            adj[v].append(u)
    return adj

def refinamento_cor(adj):
    cor = [len(viz) for viz in adj]
    while True:
        nova_cor = []
        mapa = {}
        for i in range(len(adj)):
            chave = (cor[i], tuple(sorted(cor[j] for j in adj[i])))
            if chave not in mapa:
                mapa[chave] = len(mapa)
            nova_cor.append(mapa[chave])
        if nova_cor == cor:
            return cor
        cor = nova_cor

def sao_isomorfos(g1, g2):
    a1 = incidencia_para_lista(g1)
    a2 = incidencia_para_lista(g2)
    return sorted(refinamento_cor(a1)) == sorted(refinamento_cor(a2))

def main():
    instancias = ler_instancias('Base_dados_Problema_Grafos_Isomorficos.txt')
```

```
    print("  |V| +++/--- CPU time")
    for i, (n, g1, g2) in enumerate(instancias, 1):
        ini = time.process_time()
        r = sao_isomorfos(g1, g2)
        print(f"{i}) n={n} {'+++' if r else '---'} {time.process_time() - ini:.3f}")

if __name__ == "__main__":
    main()
```