

## Ata da questão 1.1.22

Determine which pair of graphs below are isomorphic, presenting the proof by testing the smallest possible number of pairs.

Representamos os grafos através de matrizes de adjacência:

1° grafo

```
0 1 1 0 0 1 1
1 0 1 1 0 0 1
1 1 0 1 1 0 0
0 1 1 0 1 1 0
0 0 1 1 0 1 1
1 0 0 1 1 0 1
1 1 0 0 1 1 0
```

2° grafo

```
0 1 0 1 0 1 1
1 0 1 1 0 0 1
0 1 0 1 1 1 0
1 1 1 0 1 0 0
0 0 1 1 0 1 1
1 0 1 0 1 0 1
0 1 1 0 1 1 0
```

3° grafo

```
0 1 1 1 0 0 1
1 0 1 0 1 1 0
1 1 0 1 0 0 1
1 0 1 0 1 1 0
0 1 0 1 0 1 1
0 1 0 1 1 0 1
1 0 1 0 1 1 0
```

4° grafo

```
0 1 1 0 0 1 1
1 0 1 0 0 1 1
1 1 0 1 1 0 0
0 0 1 0 1 1 1
0 0 1 1 0 1 1
1 1 0 1 1 0 0
1 1 0 1 1 0 0
```

5° grafo

```
0 1 0 1 1 0 1
1 0 1 0 1 1 0
0 1 0 1 0 1 1
1 0 1 0 1 0 1
```

```
1 1 0 1 0 1 0
0 1 1 0 1 0 1
1 0 1 1 0 1 0
```

Determinamos os isomorfismos através de um programa.  
O código fonte do programa segue no fim do documento.

A entrada do programa é o número 7 (7 vértices por grafo) seguido das matrizes de adjacência acima (excluindo a linha que identifica o número de cada grafo).

O programa concluiu que os grafos 1, 2 e 5 são isomórficos e os grafos 3 e 4 são isomórficos.

## Código Fonte:

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

#define QUANTIDADE_GRAFOS 3

typedef vector< vector<int> > Matrix;

typedef pair<int, int> PII;

bool permTest(int k, const Matrix &g, const Matrix &h, vector<PII>
&gs, vector<PII> &hs) {
    const int n = g.size();
    if (k >= n) return true;
    for (int i = k; i < n && hs[i].first == gs[k].first; ++i) {
        swap(hs[i], hs[k]);
        int u = gs[k].second, v = hs[k].second;
        for (int j = 0; j <= k; ++j) {
            if (g[u][gs[j].second] != h[v][hs[j].second]) goto esc;
            if (g[gs[j].second][u] != h[hs[j].second][v]) goto esc;
        }
        if (permTest(k+1, g, h, gs, hs)) return true;
    esc:
        swap(hs[i], hs[k]);
    }
    return false;
}

bool isomorphism(const Matrix &g, const Matrix &h) {
    const int n = g.size();
    if (h.size() != n) return false;
    vector<PII> gs(n), hs(n);
```

```

    for(int i=0; i<n; ++i) {
        gs[i].second = hs[i].second = i;
        for(int j=0; j<n; ++j) {
            gs[i].first += g[i][j];
            hs[j].first += h[i][j];
        }
    }
    sort(gs.begin(), gs.end()); sort(hs.begin(), hs.end());
    for(int i=0; i<n; ++i) if (gs[i].first != hs[i].first) return
false;
    return permTest(0, g, h, gs, hs);
}

int main() {
    int n;
    cout << "Digite ou jogue um arquivo na entrada contendo:\n0
numero de vertices, seguido pela matriz de adjacencia do grafo 1\nE
em seguida a matriz de adjacencia do grafo 2, respeitando o numero de
vertices." << endl;
    cin >> n;
    vector<Matrix> vm(QUANTIDADE_GRAFOS, Matrix(n, vector<int>(n)));

    for(int k=0; k<QUANTIDADE_GRAFOS; k++) for(int i=0; i<n; ++i)
for(int j=0; j<n; ++j){
        cin >> vm[k][i][j];
    }

    for(int k=0; k<QUANTIDADE_GRAFOS; k++){
        cout << "#" << k+1 << endl;
        for(int i=0; i<n; ++i){
            for(int j=0; j<n; ++j){
                cout << vm[k][i][j] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }

    for(int a=0; a<QUANTIDADE_GRAFOS; a++) for(int b=a+1;
b<QUANTIDADE_GRAFOS; b++){

        if(isomorphism(vm[a], vm[b])) cout << a+1 << " e " << b+1 <<
" SAO ISOMORFICOS" << endl;
        else cout << a+1 << " e " << b+1 << " NAO SAO ISOMORFICOS" <<
endl;

    }
}

```