

# ÖDEV 2 İZOMORFİZM

KIVANÇ ADIGÜZEL

191180003

## Ödev 2: uygulama (bonus)

Verilecek iki farklı komşuluk matrisine göre grafların izomorfik olup olmadığını bulan program kodunu istediğiniz programlama dilini kullanarak hazırlayınız.

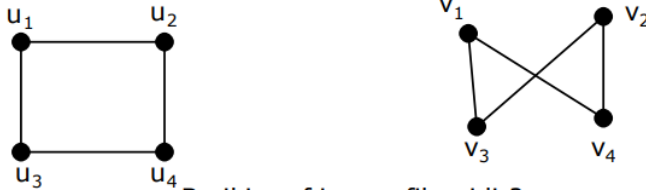
Not: Temel fonksiyonlar hariç; hazır fonksiyonların kullanılmamasına özen gösteriniz.

Öncelikle iki grafin izomorfik olup olmadığını nelere bağlıdır?

- Kenar sayıları aynı olmalıdır.
- Düğüm sayıları aynı olmalıdır.
- Düğüm dereceleri aynı olmalıdır.
- Düğüm arasındaki ilişkiyi gösteren matrisler aynı olmalıdır.

Bu matrislerdeki benzerlik satır ve sütunlardaki yer değişikliği ile de sağlanabilmektedir.

Algoritmamızı kurarken bu kuralları göz önünde bulundurmalıyız. (Direkt olarak sadece komşuluk matrisini göz önünde bulundurmak yeterli olmayacaktır.)



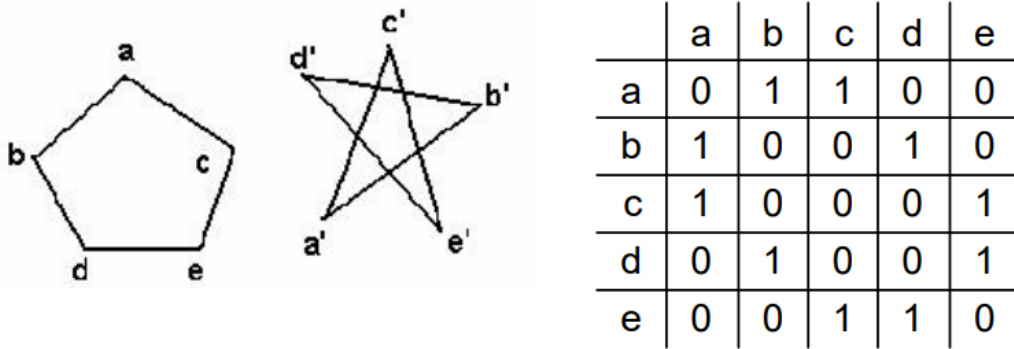
Bu iki graf izomorfik midir?

- İkisinde de 4 düğüm var
- İkisinde de 4 kenar var
- Her düğümün derecesi de 2

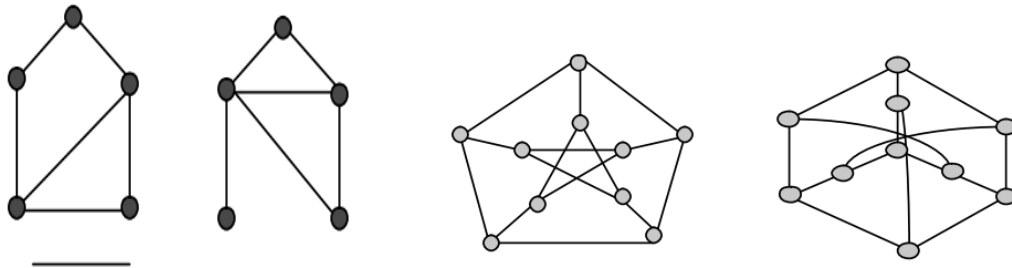
Dolayısıyla bu iki graf izomorfiktir!

	u1	u2	u3	u4	dir	v1	v2	v3	v4
u1	0	1	1	0	v1	0	0	1	1
u2	1	0	0	1	v2	0	0	1	1
u3	1	0	0	1	v3	1	1	0	0
u4	0	1	1	0	v4	1	1	0	0

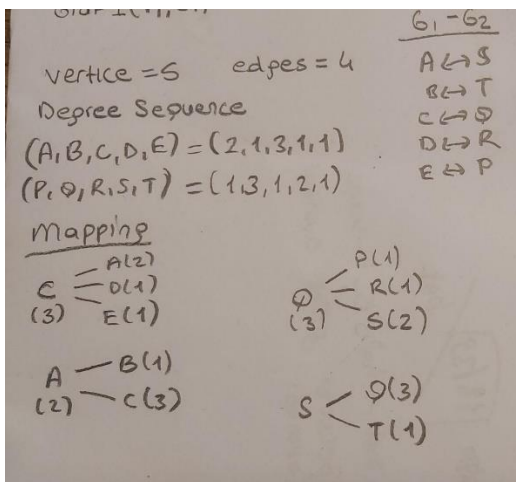
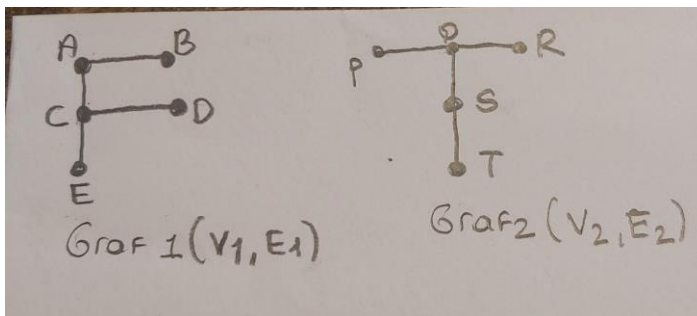
U2 ve U4 satır sütunlar yer değiştirdiğini görebiliriz.



Bu iki grafin da aynı şekilde izomorfik olduğunu söyleyebiliriz.



Bu graflar izomorfik değildir.



Algoritma kurmak için gerekli olan adımları ve bilgileri elle yazdım.

A-S, B-T, C-P, D-R, E-P

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	0	0
C	1	0	0	1	1
D	0	0	1	0	0
E	0	0	1	0	0

	S	T	Q	R	P
S	0	1	1	0	0
T	1	0	0	0	0
Q	1	0	0	1	1
R	0	0	1	0	0
P	0	0	1	0	0

Komşuluk Matrisi

Matrislerin boyutunu kullanıcıya girdirip sonra izomorfiklik durumu kontrol edilecek 2 matrisin değerlerini teker teker alan([0][0], [0][1]....) ve bunları satır toplamı yapıp yukarıda belirttiğim izomorfizm şartlarını sağlayıp sağlamama durumuna göre izomorfiklik kontrolü yapan bir algoritma tasarlanabilir.

```

Run: untitled36 x
C:\Users\Pro\CLionProjects\untitled36\cmake-build-debug\untitled36.exe
Please Enter Number of r and c : 3 3
Please Enter the Matrix Row and Column Elements
1 1 0
1 0 1
0 1 1
The Sum of Elements of a Rows in a Matrix = 2
The Sum of Elements of a Rows in a Matrix = 2
The Sum of Elements of a Rows in a Matrix = 2
Please Enter Number of 2nd r and c : 3 3
Please Enter the Matrix Row and Column Elements
1 0 1
0 1 1
1 1 0
The Sum of Elements of a Rows in a Matrix = 2
The Sum of Elements of a Rows in a Matrix = 2
The Sum of Elements of a Rows in a Matrix = 2

Process finished with exit code 0

```

```

#include<stdio.h>

void addRows(int arr[10][10], int i, int j)
{
    int rows, columns;
    for(rows = 0; rows < i; rows++)
    {
        int Sum = 0;
        for(columns = 0; columns < j; columns++)
        {
            Sum = Sum + arr[rows][columns];
        }
        printf("The Sum of Elements of a Rows in a Matrix =  %d \n", Sum );
    }
}

int main()
{
    int i, j, r, c, i2, j2, r2, c2, a[10][10], b[10][10];
    printf("Please Enter Number of r and c  :  ");
    scanf("%d %d", &i, &j);

    printf("Please Enter the Matrix Row and Column Elements \n");
    for(r = 0; r < i; r++)
    {
        for(c = 0; c < j; c++)
        {
            scanf("%d", &a[r][c]);
        }
    }

    addRows(a, i, j);

    printf("Please Enter Number of 2nd r and c  :  ");
    scanf("%d %d", &i2, &j2);

    printf("Please Enter the Matrix Row and Column Elements \n");
    for(r2 = 0; r2 < i; r2++)
    {
        for(c2 = 0; c2 < j; c2++)
        {
            scanf("%d", &b[r2][c2]);
        }
    }
    addRows(b, i2, j2);
    return 0;
}

```

Yukarda belirttiğim gibi toplamlara bakılıp izomorfizm hakkında yorum getirilebilir.

(Şu haliyle eksik fakat algoritmayı doğru kurduğumu düşünüyorum,yukarıdaki şartlar kullanıcıdan teker teker alınıp onların eşitlik durumuna göre de yorum getirilebilir,daha garanti ama uzun bir çözüm olacaktır)