

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

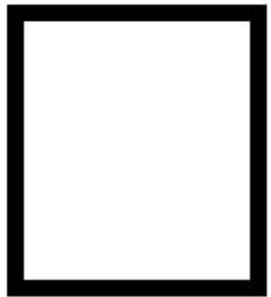
Эйлеров граф

(Эйлеров цикл, Эйлеров путь)

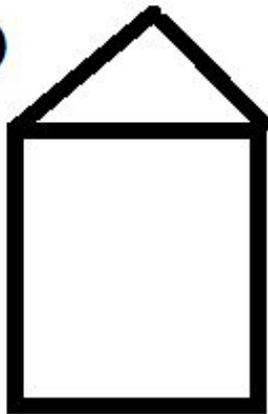
Старший преподаватель
кафедры теоретической кибернетики
Хадиев Р.М.

Можно ли не отрывая руки
нарисовать?

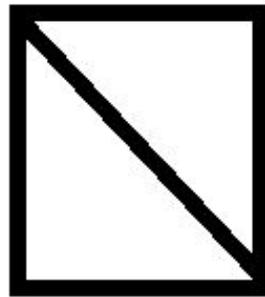
1)



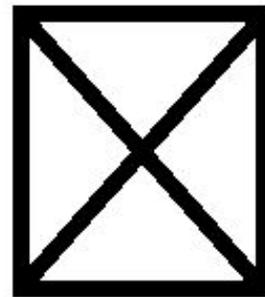
2)



3)

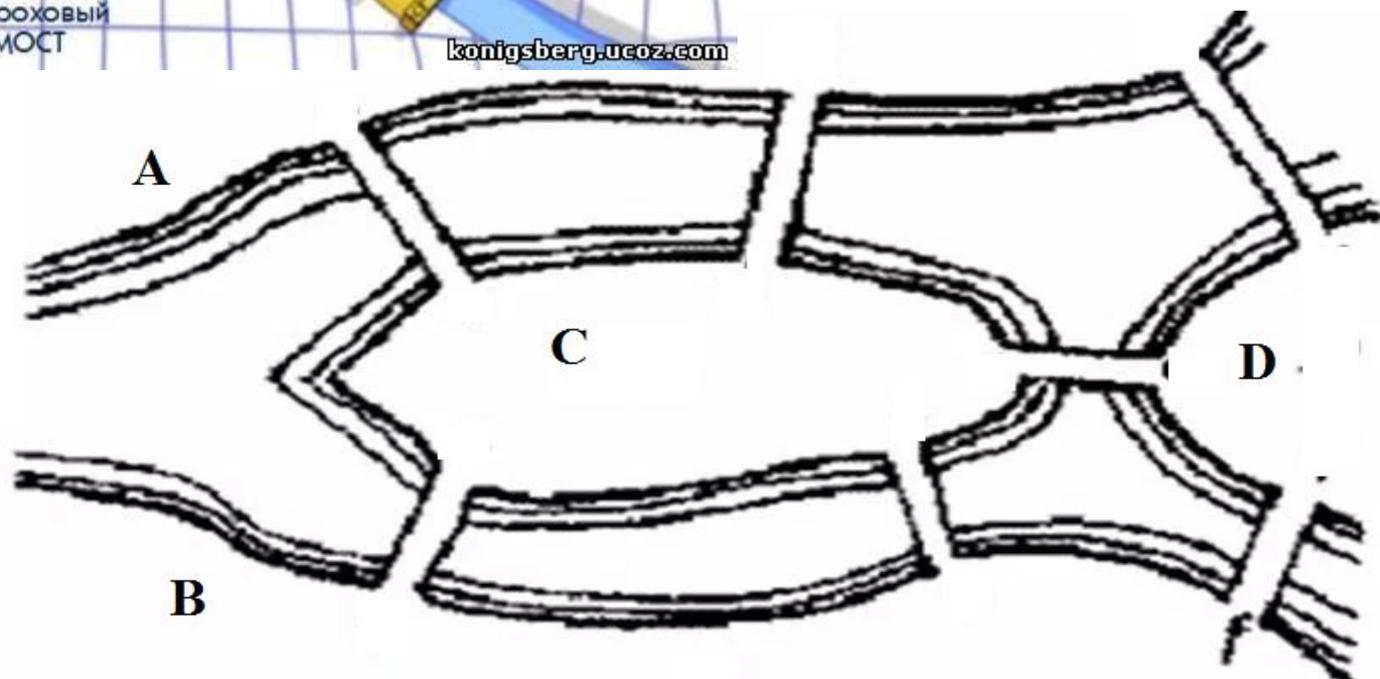
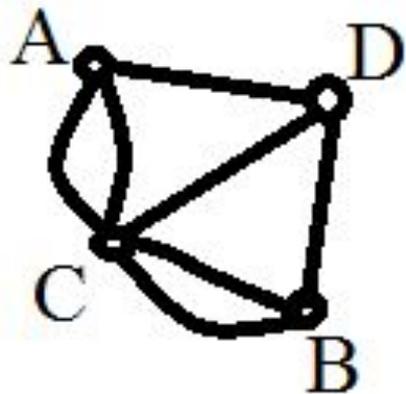


4)



5)





Можно ли обойти все Кенигсбергские мосты, проходя только один раз через каждый из этих мостов?

Свойства вершин Эйлера графа

средняя
точка
пути



$$K_{\text{выходов}} + K_{\text{входов}} = 2K$$

начальная
точка
пути



$$K_{\text{выходов}} + (K-1)_{\text{входов}} = 2K-1$$

конечная
точка
пути



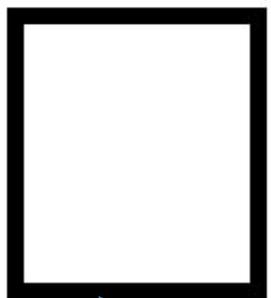
$$K_{\text{выходов}} + (K+1)_{\text{входов}} = 2K+1$$

1-ое свойство Эйлеровых графов

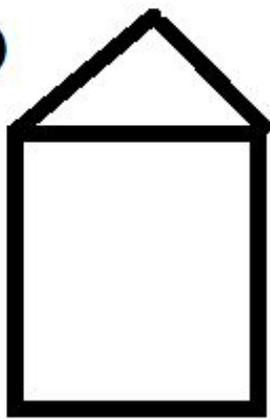
- В Эйлеровом графе число вершин с нечетной степенью равно 0 (условие существования эйлерова цикла)
- В полуэйлеровом графе число вершин с нечетной степенью равно 2 (условие существования эйлерова пути в графе)

Наши примеры

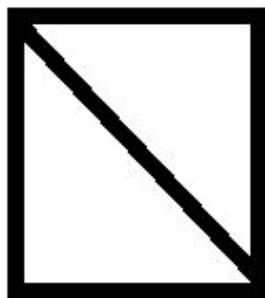
1)



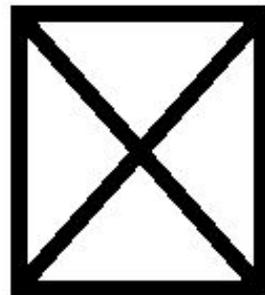
2)



3)



4)



5)



Эйлеров цикл

Эйлеров путь

Структура данных

```
int i,j,  
    n, // число вершин  
    G[100][100], //G[i][j]=1 – наличие моста  
    R[100], // степень вершины – число мостов  
cin >> n;  
// ввод данных  
for (i=1;i<=n; i++)  
    for (j=1;j<=n; j++)  
        cin >> G[i][j]
```

Подсчет степеней

```
for (i=1;i<=n;i++) {  
    R[i]=0;  
    for (j=1;j<=n;j++)  
        R[i]+=G[i][j];  
}  
  
int k=0; // число вершин с нечетной  
СТЕПЕНЬЮ  
  
for (i=1;i<=n;i++) k+=R[i]%2;
```

Выполнение первого свойства

```
if (k==0) cout << “возможно эйлеров цикл”;  
else if (k==2) cout << “возможно эйлеров путь”;  
else {  
    cout << “не эйлеров граф”;  
    return 0;  
}
```

2-ое свойство – связность графа



Пример не эйлерова графа с четными степенями вершин, но не связанного.

2-е СВОЙСТВО СВЯЗАННОСТИ

```
int Q[100]={1}; // Выявление компонент
// связности (КС). 1 – не связанная
вершина
for (i=1;i<=n;i++)
    if (R[i]==0) Q[i]=0; // изолированная вершина
i=1;
while (Q[i]==0 & i<=n) i++;
if (i>n) { cout << “вырожденный граф”; return 0;}
```

2-ое СВОЙСТВО СВЯЗАННОСТИ

```
int p[100],
    m=1; // число элементов КС
    a=1; // анализируемый элемент КС
    P[1]=i; // первый элемент КС
while (a<=m) {
    for (i=1;i<=n;i++)
        if (Q[i]==1 & G[i][P[a]]==1) {
            m++; // включение i в компоненту связности
            P[m]=i;
            Q[i]=0; // исключение i из дальнейшего
            рассмотрения
        }
    a++; // переход
}
```

2-ое СВОЙСТВО СВЯЗАННОСТИ

```
int z=0; //число нерасмотренных
        // островов с мостами
for ( int i=1; i<=n; i++)
    z+=Q[i];
if (z>0) cout << "Не эйлеров граф";
else if (k==0) cout << "Эйлеров граф";
else cout << "Полуэйлеров граф";
```