

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский Государственный технический университет”  
Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине “Математические основы интеллектуальных систем”

Тема: “Нахождение эйлеровых и гамильтоновых циклов в неориентированном графе”

**Выполнил:**

Студент 2 курса  
Группы ИИ-21  
Кирилович А. А.

**Проверил:**

Козинский А. А.

**Цель:** научиться находить эйлеровы и гамильтовы циклы в графе.

**Ход работы:**

### Задание 1

Написать программу нахождения эйлерова цикла в графе. Результатом программы является последовательность вершин в цикле. Предусмотреть проверку критерия на наличия эйлерова цикла в графе.

Граф:

(1,2),(1,3),(2,3),(2,4),(2,5),(3,4),(3,6),(5,6)

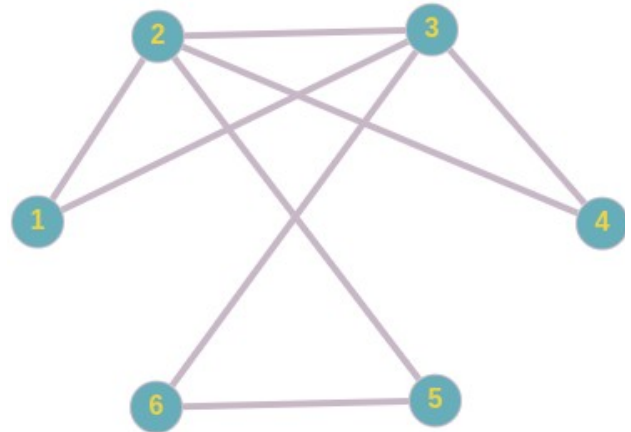
```
#include <iostream>
#include "graph_LIB.hh"

int main() {
    couple_to_matrix c;
    std::string file_path = "connections.txt";
    std::vector<int> nodes = c.reading_file(file_path);
    int max_node = c.count_of_nodes(nodes);
    std::vector<std::vector<int>> adjacencyMatrix = c.adjacency(nodes, max_node);

    // first requirement
    alg search;
    if (search.conCompBFS(adjacencyMatrix, max_node) != 1) {
        std::cout << "The graph not coherent";
        exit(0);
    }

    // second requirement
    std::vector<int> degrees(max_node);
    for (int i = 0; i < max_node; i++) {
        for (int j = 0; j < max_node; j++) {
            if (adjacencyMatrix[i][j]) {
                degrees[i]++;
            }
        }
    }
    for (int i = 0; i < max_node; i++) {
        if (degrees[i] % 2 != 0) {
            std::cout << "The graph not Eulerian";
            exit(0);
        }
    }

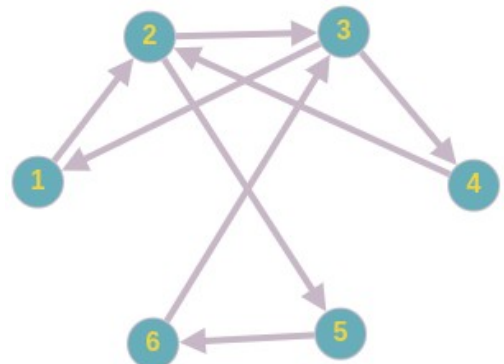
    // find the Eulerian cycle
    std::vector<int> cycle;
    cycle = search.findEulerianCycle(adjacencyMatrix, max_node);
    std::cout << "The cycle is: ";
    for (int i = cycle.size() - 1; i >= 0; i--) {
        std::cout << cycle[i] + 1;
    }
}
```



The cycle is: 123425631

Реализация функции findEulerianCycle из файла Graph\_LIB:

```
std::vector<int> findEulerianCycle(std::vector<std::vector<int>> &adjacencyMatrix, int max_node) {
    std::vector<int> cycle;
    int start = 0;
    stack s;
    s.push(start);
    while (!s.is_empty()) {
        start = s.front();
        bool found_edge = false;
        for (int i = 0; i < max_node; i++) {
            if (adjacencyMatrix[start][i]) {
                s.push(i);
                adjacencyMatrix[start][i] = 0;
                adjacencyMatrix[i][start] = 0;
                found_edge = true;
                break;
            }
        }
    }
}
```



```

    }
    }
    if (!found_edge) {
        s.pop();
        cycle.push_back(start);
    }
}
return cycle;
}

```

## Задание 2

Написать программу нахождения гамильтонова цикла в графе. Результат программы – последовательность вершин цикла.

Граф:

(a,b),(a,c),(a,d),(b,e),(b,f),(c,d),(c,e),(e,f) →  
→ (1,2),(1,3),(1,4),(2,5),(2,6),(3,4),(3,5),(5,6)

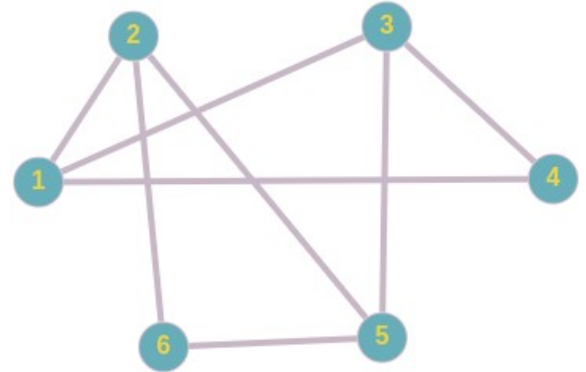
```

#include <iostream>
#include "graph_LIB.hh"

int main() {
    couple_to_matrix c;
    std::string file_path = "connections.txt";
    std::vector<int> nodes = c.reading_file(file_path);
    int max_node = c.count_of_nodes(nodes);
    std::vector<std::vector<int>> adjacencyMatrix =
c.adjancy(nodes, max_node);

    // find Hamiltonian Cycle
    queue q;
    alg search;
    q = search.findHamiltonianCycle(adjacencyMatrix, max_node);
    q.print();
}

```



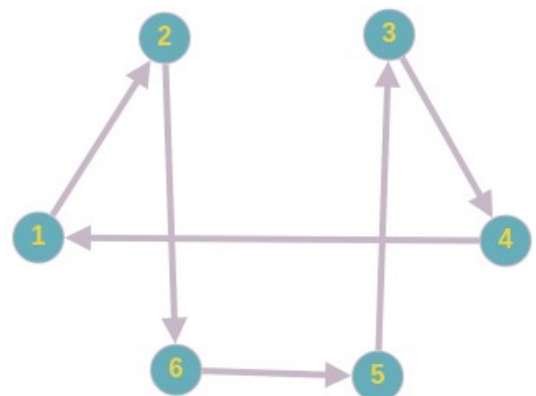
015423

Реализация функции findHamiltonianCycle из файла Graph\_LIB:

```

queue findHamiltonianCycle(std::vector<std::vector<int>> &adjacencyMatrix, int max_node) {
    queue q;
    for (int i = 0; i < max_node; i++) {
        q.push(i);
    }
    int i;
    for (int k = 0; k < max_node * (max_node - 1); k++) {
        if (adjacencyMatrix[q[0]][q[1]] != 1) {
            i = 1;
            while (adjacencyMatrix[q[0]][q[i]] != 1 || adjacencyMatrix[q[1]][q[i + 1]] != 1) {
                i++;
            }
            for (int j = 0; 1 + j < i - j; j++) {
                q.swap(1 + j, i - j);
            }
        }
        q.push(q.front());
        q.pop();
    }
    return q;
}

```



**Вывод:** в ходе лабораторной работы я научился находить эйлеровы и гамильтовы циклы в графах.