

# Разработка программы решения теоретико-графовой задачи

## Перечень теоретико-графовых задач

### 1. Определить вид графа:

1. (1) Дерево (нг, ог)
2. (1) Ациклический граф (нг, ог)
3. (1) Полуэйлеров/эйлеров граф (нг)
4. (3) Гамильтонов граф (нг)
5. (0) Полный граф (нг)
6. (2) Связный граф (нг)
7. (3) Сильно-связный граф (ог)
8. (2) Двусвязный граф
9. (2) Двудольный неориентированный граф
10. (2) Регулярный, реберно-регулярный неориентированный граф
11. (2) Симметричный, антисимметричный, частично симметричный орграф
12. (2) Транзитивный, антитранзитивный, частично транзитивный орграф
13. (2) Рефлексивный, антирефлексивный, частично рефлексивный орграф
14. (2) Функциональный, контрафункциональный орграф
15. (1) Вычерчиваемый граф
16. (2) Односторонне связный орграф
17. (4) Кактус
18. (1) Турнир, транзитивный турнир (ог)
19. (1) Граф Бержа (ог)
20. (3) Граф Паппа (нг)

21. (4) Планарный граф (нг, ог)
22. (3) Транзитируемый граф (нг)

## **2. Определить числовую характеристику графа:**

1. (2) Радиус (нг, ог, внг, вог)
2. (2) Диаметр (нг, ог, внг, вог)
3. (2) Средний диаметр (нг, ог, внг, вог)
4. (1) Полустепени захода/исхода и средние полустепени всех вершин в орграфе
5. (1) Минимальную степень/среднюю степень/максимальную степень ребра в неориентированном графе
6. (4) Число вершинной связности (нг, ог)
7. (4) Число рёберной связности (нг, ог)
8. (2) Среднее и максимальное расстояние между центральными вершинами неориентированного графа
9. (3) Число хорд неориентированного графа
10. (2) Минимальное и среднее расстояние между периферийными вершинами неориентированного графа
11. (0) Цикломатическое число неориентированного графа
12. (3) Окружение орграфа
13. (3) Обхват орграфа
14. (2) Число компонентов связности неориентированного графа
15. (3) Число Хадwigера для неориентированного графа
16. (5) Определить толщину неориентированного графа
17. (4) Индекс компонент относительно простой цепи в неориентированном графе

## **3. Построение графа:**

1. (4) Сгенерировать клетку указанного обхвата
2. (?) Найти граф, являющийся пересечением множества всех диаметральных цепей неориентированного графа

3. (?) Найти граф, являющийся объединением всех радиусов графа
4. (?) Найти граф, являющийся объединением множества всех диаметров графа
5. (?) Найти граф, являющийся пересечением множества всех гамильтоновых циклов графа
6. (?) Найти граф, являющийся объединением множества всех гамильтоновых циклов графа

#### **4. Операции над графами:**

1. (2) Найти декартово произведение двух неориентированных графов
2. (2) Найти декартову сумму двух неориентированных графов
3. (2) Найти прямое (тензорное) произведение двух неориентированных графов
4. (2) Найти сильное произведение двух неориентированных графов
5. (2) Найти композицию двух неориентированных графов
6. (2) Найти модульное произведение двух неориентированных графов
7. (2) Найти большое модульное произведение двух неориентированных графов
8. (2) Найти объединение множества неориентированных графов
9. (2) Найти пересечение множества неориентированных графов
10. (2) Найти дополнение и фактор-дополнение неориентированного графа
11. (2) Найти граф инциденций неориентированного графа
12. (2) Найти реберный граф для неориентированного графа
13. (2) Найти граф смежностей для неориентированного графа
14. (2) Найти тотальный граф для неориентированного графа
15. (2) Найти граф замыкания неориентированного графа
16. (4) Найти граф конденсации для орграфа
17. (4) Найти граф каркасов для неориентированного графа

#### **5. Поиск в графе:**

1. (2) Определить эксцентриситет каждой вершины в неориентированном графе
2. (3) Найти эйлеров цикл в графе (нг, ог)
3. (3) Найти гамильтонов цикл (нг, ог)
4. (3) Найти компоненты связности в неориентированном графе
5. (3) Найти сильные компоненты связности в орграфе
6. (0) Найти вершины с указанной степенью
7. (3) Найти минимальный остов в неориентированном графа
8. (3) Найти доли неориентированного графа
9. (0) Найти тупики/антитупики в ориентированном графе
10. (3) Найти максимальный простой разрез (нг, ог, внг, вог)
11. (3) Найти минимальный простой разрез (нг, ог, внг, вог)
12. (?) Найти все пары вершин с указанным расстоянием (нг, ог, внг, вог)
13. (?) Найти множество рёбер, удаление которых приводит к увеличению числа компонентов связности ориентированного графа
14. (3) Найти точки сочленения неориентированного графа
15. (3) Найти мосты в неориентированном графе
16. (5) Найти множество вершин, удаление которых приводит к увеличению числа компонентов связности ориентированного графа
17. (3) Найти циклы указанной длины (нг, ог, внг, вог)
18. (3) Найти максимальный путь между заданными вершинами (нг, ог, внг, вог)
19. (?) Нахождение критических путей во взвешенном неориентированном графе
20. (?) Найти множество вершин, удаление которых приводит к увеличению числа компонентов связности неориентированного графа
21. (3) Найти простые цепи указанной длины (нг, ог, внг, вог)
22. (?) Найти вершины с указанной полустепенью
23. (1) Найти все доминирующие вершины (нг, ог, внг, вог)

24. (?) Найти центры графа
25. (?) Найти рёбра с указанной степенью
26. (2) Найти все периферийные вершины (нг, ог, внг, вог)
27. (?) Найти множество рёбер, удаление которых приводит к увеличению числа компонентов связности неориентированного графа
28. (?) Найти звёзды с заданным числом листьев
29. (4) Найти дерево кратчайших путей (нг, ог, внг, вог)
30. (?) Определить рёбра и их степени в неориентированном мультиграфе
31. (?) Найти  $n$ -фактор для указанного графа
32. (5) Поиск подграфов в неориентированном графе, изоморфных графу-образцу
33. (3) Сформировать множество различных суграфов неориентированного графа
34. (3) Сформировать множество различных подграфов неориентированного графа

## **6. Приведение графа к указанному виду:**

1. (3) Найти минимальное множество вершин неориентированного графа, удаление которых позволяет сделать его деревом
2. (3) Найти минимальное множество рёбер неориентированного графа, удаление которых позволяет сделать его деревом
3. (5) Найти минимальное множество вершин неориентированного графа, удаление которых позволяет сделать его планарным
4. (5) Найти минимальное множество рёбер графа, удаление которых позволяет сделать его планарным

**Вариант задания: 1.2 мс**

Граф — комбинация набора вершин и набора рёбер.

Вершина — точка графа.

Ребро — путь от одной вершины к другой.

Граф, в котором все рёбра неориентированные(двухсторонние), также называют неориентированным, а граф с ориентированными рёбрами(односторонними) — ориентированным.

Путём в графе называется последовательность вершин, каждая из которых соединена со следующей ребром.

Путь, который проходит через какую-либо вершину более одного раза называют сложным путём.

Если первая вершина пути совпадает с последней, то такой путь называют циклом.

Матрица смежности — это вид представления графа в виде матрицы, когда пересечение столбцов и строк задаёт дуги. Граф из  $N \times N$  вершин задаётся матрицей (двумерным массивом)  $N \times N$ , в которой  $g[i][j]$  - логическое значение, true или false, обозначающее, существует ли ребро из вершины  $i$  в вершину  $j$ .

Ациклический граф — граф, в котором отсутствуют направленные циклы, но могут быть «параллельные» пути, выходящие из одного узла и разными путями приходящие в конечный узел.

## Алгоритм

1. Просим пользователя ввести количество вершин (*"Enter number of vertexes = "*) и количество выполнения цикла for (*"Enter number of cycles = "*), в котором происходит создание пути между вершинами по средствам введения двух рёбер (*"Enter edge u(" << i + 1 << ") = "* и *"Enter edge v(" << i + 1 << ") = "*, где  $i$  – int переменная цикла). Из двух введенных рёбер получаем два значения

для двумерного массива bool переменной (graph[u][v]).

2. В следующем цикле for выполняется поиск связей между вершинами. Если с вершиной связан один путь, то текущий граф становится ациклическим графом (isAcyclic = true). В другом случае значение isAcyclic, значение bool переменной которого изначально было false, не меняется.
3. Выводятся вершины и пути к этим вершинам согласно введенным рёбрам и количеству вершин (с << " edges adjacent to vertex " << i + 1, где с - int переменная количества путей, i – int переменная вершин).
4. Выводится "Current graph is acyclic" при isAcyclic = true, иначе "Current graph is not acyclic".

### Тесты:

```
Enter number of vertexes = 5
Enter number of cycles = 5
Enter edge u(1) = 1
Enter edge v(1) = 3
Enter edge u(2) = 5
Enter edge v(2) = 2
Enter edge u(3) = 1
Enter edge v(3) = 6
Enter edge u(4) = 4
Enter edge v(4) = 1
Enter edge u(5) = 4
Enter edge v(5) = 5
2 edges adjacent to vertex 1
1 edges adjacent to vertex 2
1 edges adjacent to vertex 3
2 edges adjacent to vertex 4
2 edges adjacent to vertex 5
Current graph is not acyclic
D:\Games_Install\Уроки\PiOIwI
Чтобы автоматически закрывать
Нажмите любую клавишу, чтобы
```

```
Enter number of vertexes = 6
Enter number of cycles = 6
Enter edge u(1) = 1
Enter edge v(1) = 3
Enter edge u(2) = 4
Enter edge v(2) = 2
Enter edge u(3) = 5
Enter edge v(3) = 6
Enter edge u(4) = 1
Enter edge v(4) = 7
Enter edge u(5) = 8
Enter edge v(5) = 2
Enter edge u(6) = 5
Enter edge v(6) = 9
1 edges adjacent to vertex 1
1 edges adjacent to vertex 2
1 edges adjacent to vertex 3
1 edges adjacent to vertex 4
1 edges adjacent to vertex 5
1 edges adjacent to vertex 6
Current graph is acyclic
```

```
Enter number of vertexes = 7
Enter number of cycles = 7
Enter edge u(1) = 8
Enter edge v(1) = 1
Enter edge u(2) = 6
Enter edge v(2) = 22
Enter edge u(3) = 5
Enter edge v(3) = 8
Enter edge u(4) = 2
Enter edge v(4) = 9
Enter edge u(5) = 4
Enter edge v(5) = 6
Enter edge u(6) = 1
Enter edge v(6) = 5
Enter edge u(7) = 11
Enter edge v(7) = 7
1 edges adjacent to vertex 1
0 edges adjacent to vertex 2
0 edges adjacent to vertex 3
1 edges adjacent to vertex 4
1 edges adjacent to vertex 5
1 edges adjacent to vertex 6
0 edges adjacent to vertex 7
Current graph is not acyclic
```

```

Enter number of vertexes = 8
Enter number of cycles = 8
Enter edge u(1) = 1
Enter edge v(1) = 3
Enter edge u(2) = 5
Enter edge v(2) = 7
Enter edge u(3) = 9
Enter edge v(3) = 2
Enter edge u(4) = 8
Enter edge v(4) = 4
Enter edge u(5) = 6
Enter edge v(5) = 2
Enter edge u(6) = 1
Enter edge v(6) = 9
Enter edge u(7) = 6
Enter edge v(7) = 12
Enter edge u(8) = 9
Enter edge v(8) = 7
1 edges adjacent to vertex 1
1 edges adjacent to vertex 2
1 edges adjacent to vertex 3
1 edges adjacent to vertex 4
1 edges adjacent to vertex 5
1 edges adjacent to vertex 6
1 edges adjacent to vertex 7
1 edges adjacent to vertex 8
Current graph is acyclic

```

```

Enter number of vertexes = 9
Enter number of cycles = 9
Enter edge u(1) = 5
Enter edge v(1) = 3
Enter edge u(2) = 7
Enter edge v(2) = 9
Enter edge u(3) = 2
Enter edge v(3) = 5
Enter edge u(4) = 1
Enter edge v(4) = 3
Enter edge u(5) = 1
Enter edge v(5) = 4
Enter edge u(6) = 6
Enter edge v(6) = 3
Enter edge u(7) = 8
Enter edge v(7) = 7
Enter edge u(8) = 10
Enter edge v(8) = 9
Enter edge u(9) = 3
Enter edge v(9) = 5
2 edges adjacent to vertex 1
1 edges adjacent to vertex 2
3 edges adjacent to vertex 3
1 edges adjacent to vertex 4
2 edges adjacent to vertex 5
1 edges adjacent to vertex 6
2 edges adjacent to vertex 7
1 edges adjacent to vertex 8
1 edges adjacent to vertex 9
Current graph is acyclic

```

## Вывод

В расчетной работе создали программу на языке C++, в которой создается граф с помощью матрицы смежности и проверяется является ли созданный граф ациклическим.

## Ресурсы:

Понятие и представление графа: матрица смежности, список смежности [Электронный ресурс]. – Режим доступа: <https://brestprog.by/topics/graphs/> – Дата доступа: 18.12.2024.

Ориентированный ациклический граф [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Ориентированный\\_ациклический\\_граф](https://ru.wikipedia.org/wiki/Ориентированный_ациклический_граф) – Дата доступа: 18.12.2024.