Задача 7. Графы

В рамках данной задачи требуется разработать приложение, которое будет решать поставленную задачу с использованием графов. В связи с этим, требуется организовать удобный пользовательский интерфейс, позволяющий конечному пользователю каким-либо образом вводить графы и выводить их на экран.
Для ввода можно использовать либо матрицу смежности и, при необходимости, массив вершин, либо описание графа в формате языка dot. Возможна комбинация обоих методов.
Выводить граф можно с использованием собственного рендеринга или, опять-таки, в формате языка dot с последующей визуализацией в программе GV.
Предусмотреть возможность сохранения графа в файл и его загрузку из файла.
При разработке класса, хранящего граф, рекомендуется использовать интерфейс Graph, описанный в демонстрационном проекте на лекциях. В качестве основы для разработки собственного приложения (организации пользовательского интерфейса ввода-вывода графов) можно использовать то же демонстрационное приложение.

Варианты:

1. Найти все вершины графа, недостижимые из заданной вершины. Вывести в виде списка и пометить при графическом выводе (рекурсивная реализация).

2. Определить вершину, удалением которой можно свести граф к дереву.

3. Определить, является ли связанным заданный граф (рекурсивная реализация).

4. Найти длину кратчайшего цикла в графе

5. Найти все циклы, не содержащие указанные вершины.

6. Найти вершину(ы) заданного графа, которая принадлежит каждому пути между двумя заданными вершинами.

7. Для заданного графа знакомств определить, кого с кем надо познакомить, чтобы граф стал удовлетворять теории об N рукопожатиях. Минимизировать количество знакомств.

8. Задана система односторонних дорог. Найти путь, соединяющий города А и Б, не проходящий через заданное множество вершин.

9. Найти город в системе двусторонних дорог, у которого сумма расстояний до любого города минимальна.

10. По системе двусторонних дорог определить, есть ли в ней город (вывести его), из которого можно добраться в любой другой менее чем за N км.

11. Задана система двусторонних дорог. Найти замкнутый путь, проходящий через каждую вершину и длиной не более N км.

12. По системе двусторонних дорог определить, можно ли, закрыв какие-либо K из них, добиться того, чтобы из города А нельзя было попасть в город Б.

13. Для заданного графа знакомств определить, удаление каких узлов / каких рёбер приведёт к тому, что граф перестанет удовлетворять теории об N рукопожатиях (рассмотреть варианты с допустимым разбиением на два несвязанный подграфа, так и с недопустимым).

14. Задана система двусторонних дорог. N-периферией называется множество городов, расстояние от которых до выделенного города больше N. Определить N-периферию для заданного города.

15. В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа ровно один раз.

16. Имеется N городов. Для каждой пары городов (I,J) можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги A(I,J). Вне городов дороги не пересекаются. Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей B[1:N,1:N], где B[I,J]=1 тогда и только тогда, когда дорогу, соединяющую города I и J, следует строить.

17. Раскрасить граф минимальным количеством цветов. Каждая вершина должна быть «окрашена» в цвет, отличный от цвета смежных вершин.

18. Вычислить число различных циклов в полном графе с n вершинами (неориентированом) и вывести их на экран.

19. Найти рёбра заданного графа, которые принадлежат каждому пути между двумя заданными вершинами.

20. Дан неориентированный граф с N вершинами без изолированных вершин. Вывести все подграфы с M ребрами

21. Вывести все возможные пути (без циклов) из точки А в точку Б в порядке увеличения общего расстояния.

22. N колец сцеплены между собой (задана матрица A(n*n), A(i,j)=1 в случае, если кольца i и j сцеплены друг с другом и A(i,j)=0 иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.

23. Для заданного графа знакомств определить число N, такое, чтобы этот граф стал удовлетворять теории об N рукопожатиях.

24. Для заданного графа знакомств найти все узлы, которые удовлетворяют теории N рукопожатий.

25. Задан набор неповторяющихся пар (Ai,Aj), Ai, Aj принадлежат множеству A={A1, A2, ..., An}. Необходимо составить цепочку максимальной длины по правилу: (Ai,Aj)+(Aj,Ak)=(Ai,Aj,Ak).

26. Для заданного графа знакомств определить, удовлетворяет ли граф теории N рукопожатий (каждый узел знаком опосредованно со всеми остальными не более чем через N-1 узлов)

27. Для заданного графа знакомств определить узлы, которые не удовлетворяют условию N рукопожатий для указанного узла.

28. Лабиринт задается матрицей смежности N*N, где C(i,j)=1, если узел i связан узлом j посредством дороги. Часть узлов назначается входами, часть - выходами. Входы и выходы задаются последовательностями узлов X(1),..,X(p) и Y(1),..,Y(k) соответственно.
Найти максимальное число людей, которых можно провести от входов до выходов таким образом, чтобы:
а) их пути не пересекались по дорогам, но могут пересекаться по узлам;
б) их пути не пересекались по узлам;

29. Определить ребро, удалением которого можно свести граф к дереву.

30. Реализовать алгоритм, который находит рёбра неориентированного графа, исключая которые граф разделяется на два подграфа, количество вершин в которых отличается не более, чем в K раз.

31. Задан граф знакомств. Между каждой парой знакомых возможны только два вида отношений: союзник и противник (обоюдно). Если считать, что: противник противника = союзник и союзник союзника = союзник, а противник союзника и союзник противника – противники, для каждого узла определить, является ли он союзником или противником для заданного узла. В случае, если невозможно однозначно установить, то (1) пометить такие узлы, как нейтральные; (2) попытаться вычислить дружественность таких узлов, где 0 – соперник, а 1 – союзник (можно перенести с интервала [0..1] на [-1..1]). Например, если А союзник Б, А союзник В, Б союзник Г и В соперник Г, то получается, что Г на 50% союзник А. Но если добавить Д, который является союзником А и Б, то Г станет союзником А на 2/3.

32. Задан граф знакомств. Между каждой парой знакомых возможны только два вида отношений: союзник и противник (обоюдно). Для заданной пары заведомо противников (прямых или опосредованных), определить, какой из узлов чью сторону займёт или примет нейтралитет, если узел может принять решение о союзничестве с одной из сторон только в том случае, когда коэффициент дружественности для этой стороны более X и разница коэффициентов составляет Y.

33. Составить программу для нахождения произвольного разбиения N студентов на M команд, численность которых отличается не более чем в K раза, если известно, что в любой команде должны быть студенты, обязательно знакомые друг с другом (Рассмотреть два варианта: с допустимым опосредованным знакомством и обязательным личным знакомством).

34. Неориентированный граф называется двудольным, если его можно раскрасить в два цвета так, что концы любого ребра будут разного цвета. Составить алгоритм проверки, является ли заданный граф двудольным (число действий не превосходит N + M).

35. Имеется N человек и прямоугольная таблица A[1:N,1:N], которая задаёт граф знакомств. Элемент A[i,j] равен 1 (имеется соответствующее ребро в графе), если человек i знаком с человеком j, A[i,j] = =A[j,i]. Можно ли разбить людей на K групп (2<=K<=N), чтобы в каждой группе были только незнакомые люди.

36. Пусть группа состоит из N человек. В ней каждый имеет (N/2) друзей и не больше K врагов. У одного из них есть книга, которую все хотели бы прочитать и потом обсудить с некоторыми из остальных.
Написать программу, которая:
1. Находит способ передачи книги таким образом, чтобы она побывала у каждого в точности один раз, переходя только от друга к другу и наконец возвратилась к своему владельцу.
2.Разбивает людей на S групп, где будет обсуждаться книга, таким образом, чтобы вместе с каждым человеком в ту же самую группу вошло не более P его врагов.
Примечание: предполагается, что S*P>=K.

37.