

КАФЕДРА АВТОМАТИЗИРОВАННЫХ  
СИСТЕМ УПРАВЛЕНИЯ

# ДИСЦИПЛИНА

« Дискретная математика »

*Тема № 3. Графы.*

*Лекция. Раскраска графов.*

# УЧЕБНЫЕ ВОПРОСЫ

1. Основные понятия. Хроматическое число
2. Алгоритмы раскраски графа

# 1. Основные понятия. Хроматическое число

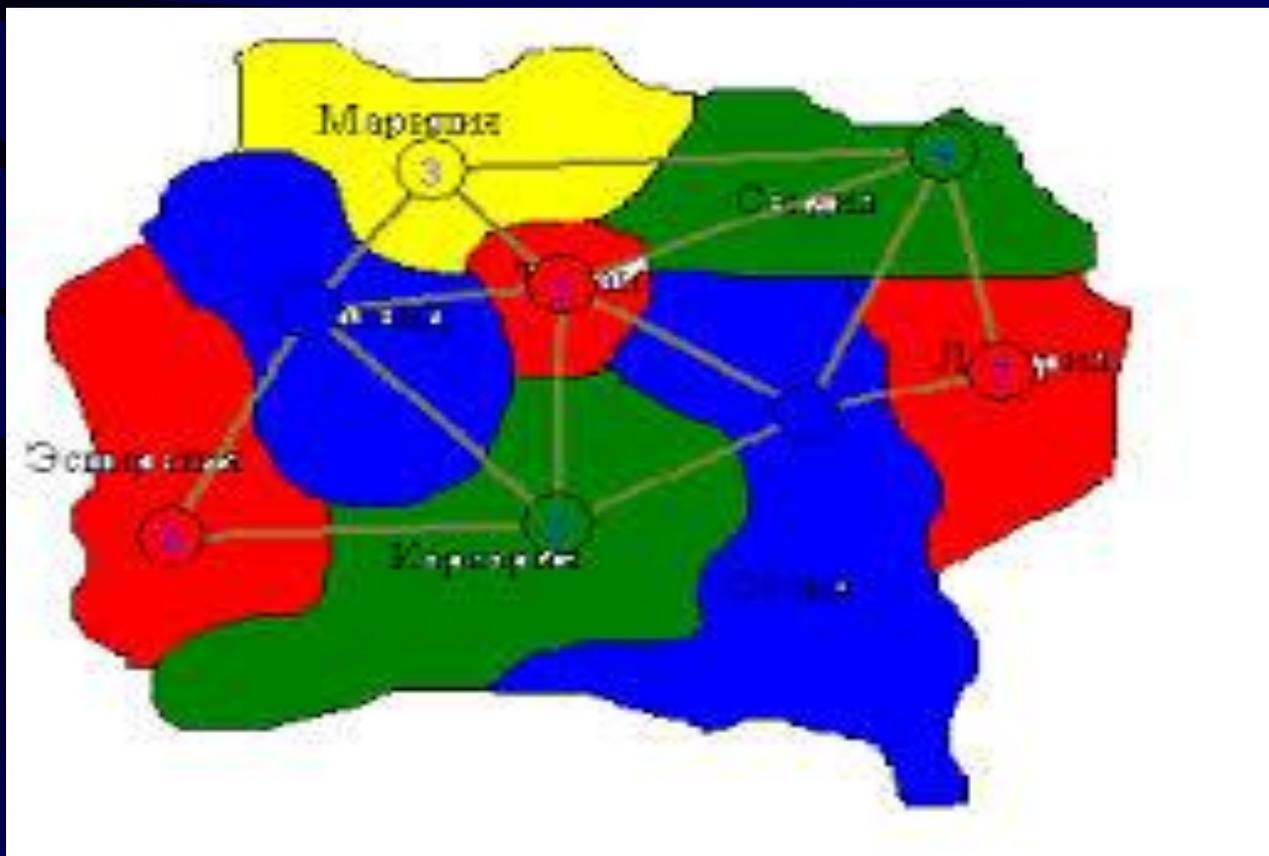
Характерным специфическим направлением теории графов является цикл задач, связанный с раскрасками графов, в котором изучаются разбиения множества вершин (ребер), обладающие определенными свойствами, например, смежные вершины (ребра) должны принадлежать различным множествам (вершины или ребра из одного множества окрашиваются одним цветом).

История возникновения теории графов связана с задачей о раскраске графов.

**Задача о четырех красках.** Формулировка задачи проста и не соответствует всей глубине и сложности проблемы: *можно ли на любой политико-административной карте раскрасить страны так, чтобы никакие две страны, имеющие общую границу, не были раскрашены одинаковой краской, и чтобы были использованы всего четыре краски.*

Уточним, что если две страны граничат по точке, то они не считаются имеющими общую границу.

Каждый многоугольный граф можно представить себе как некоторую географическую карту, где грани - это страны, а бесконечная грань- окружающий их океан. На такой карте все страны и океан раскрашиваются так, чтобы их можно было отличить друг от друга. Для этого страны с общей границей должны быть раскрашены в разные цвета. Если в нашем распоряжении имеется достаточное количество красок, это не составит никакого труда. Намного сложнее решить вопрос о наименьшем количестве красок, достаточного для такого раскрашивания стран данной карты



Задача эта приобрела известность с 1879 г., когда английский математик **Артур Кэпи** привел ее формулировку на заседании английского королевского научного общества; добавив, что не мог ее решить, хотя и размышлял над ней длительное время. С тех пор многие выдающиеся математики пробовали свои силы в решении этой задачи.

В терминах графов задача может быть поставлена следующим образом:

*Дан произвольный полностью неориентированный плоский граф  $G$ . Можно ли каждую вершину графа  $G$  раскрасить с помощью одной из четырех красок так, чтобы никакие две смежные вершины (вершины, соединенные хотя бы одним ребром) не были раскрашены в один цвет.*

*Проблема четырех красок до сих пор не решена. Удалось лишь доказать, что такую раскраску можно осуществить для всех плоских графов с числом вершин, не превосходящим 38.*

Другое применение раскраски в графах может быть использовано при:

- распределение памяти в ЭВМ;
- проектирование сетей телевизионного вещания;
- распределение частот и областей сотовых систем связи.

Задача об определении минимального числа красок, нужных для правильной раскраски графа, которую рассмотрим, является одной из первых задач теории графов. При этом рассматриваемый *граф должен быть полностью неориентированный без кратных ребер и петель.*

Раскраска вершин.

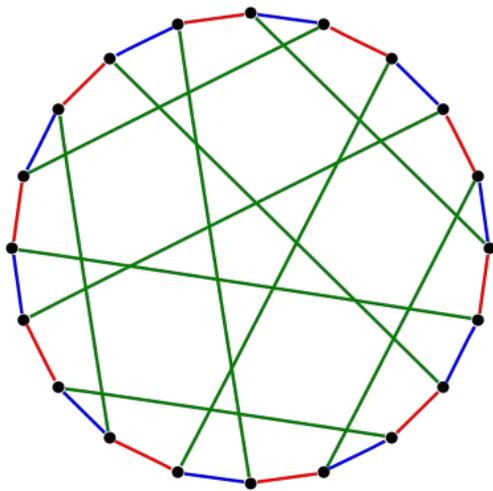
**Виды раскрасок**



Терминология, в которой метки называются цветами, происходит от раскраски политических карт. Такие метки как **красный** или **синий** используются только когда число цветов мало, обычно же подразумевается, что метки являются целыми числами  $\{1,2,3,\dots\}$ . *Раскраска с использованием  $k$  цветов называется  $k$ -раскраской.*

**Определение.** Наименьшее число цветов, необходимое для раскраски графа  $G$  называется его **хроматическим числом** и часто записывается как  $X(G)$ . ( Иногда используется  $Y(G)$ , с тех пор как  $X(G)$  обозначает Эйлерову характеристику).

Подмножество вершин, выделенных одним цветом, называется *цветовым классом*, каждый такой класс формирует независимый набор. Таким образом,  $k$ -раскраска - это то же самое, что и разделение вершин на  $k$  независимых наборов.



## Рёберная раскраска

Рисунок 1 - Рёберная раскраска графа в красный, синий и зелёный цвета.

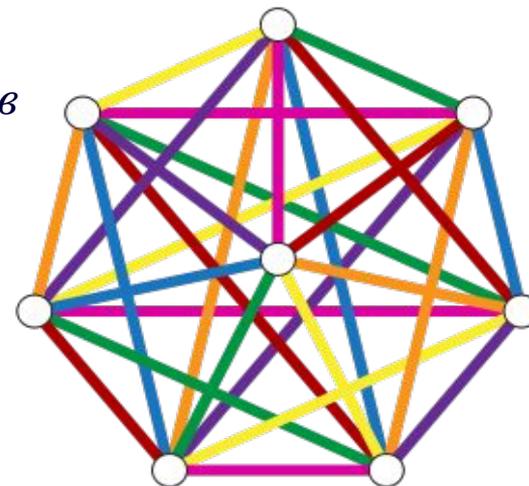
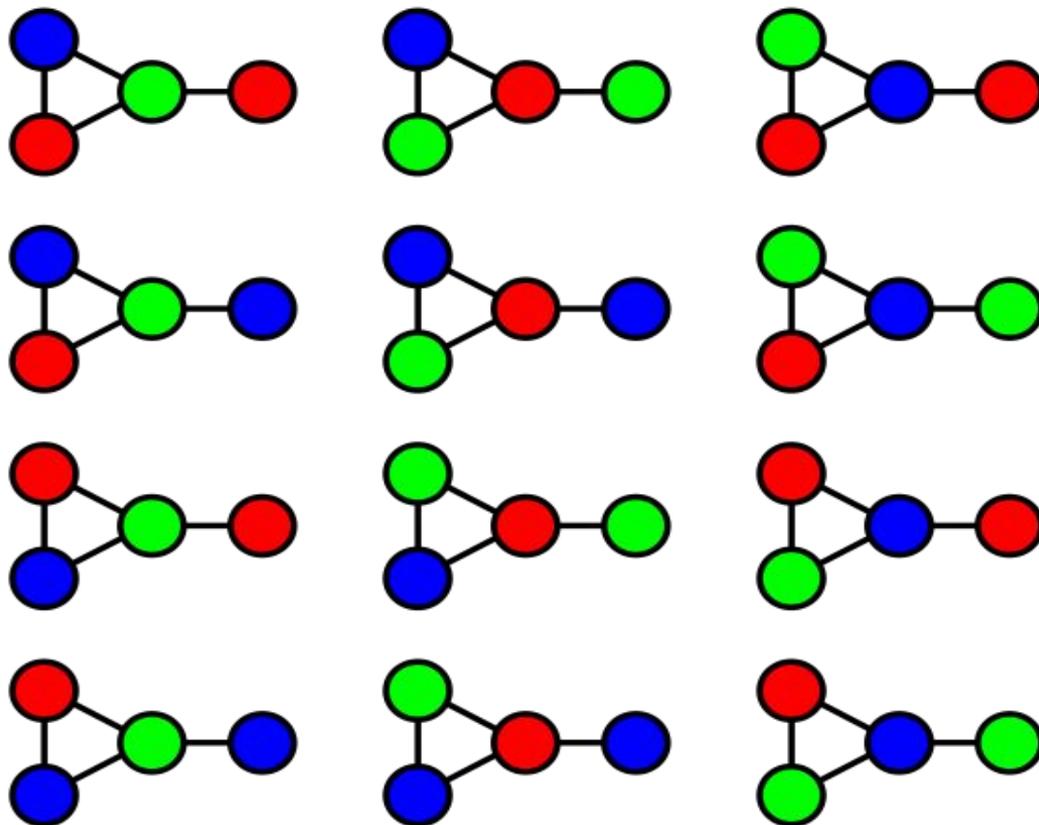
### Рёберная 3-цветная раскраска

В теории графов рёберной раскраской графа называется назначение «цветов» рёбрам графа таким образом, что никакие два сопряжённых ребра не имеют один и тот же цвет.

Рёберная раскраска — это один из видов различных типов раскраски графов. Задача рёберной раскраски задаётся вопросом, *можно ли раскрасить рёбра заданного графа максимум в  $k$  различных цветов для заданного значения  $k$  или для минимального возможного числа цветов.* **Минимальное требуемое число цветов для раскраски рёбер заданного графа называется хроматическим индексом графа.**

Например, рёбра графа на иллюстрации можно раскрасить в три цвета, но нельзя раскрасить в два, так что граф имеет хроматический индекс три.

*Пример - Построение раскраски графа  $K_8$  в 7 цветов*



*Рисунок 2 - Граф может быть раскрашен 3 цветами 12 способами*

## Хроматический многочлен

*Хроматический многочлен* считает число возможных вариантов раскраски графа с использованием не более чем заданного числа цветов.

*Например*, граф на рисунке 2 может быть раскрашен 12 способами с использованием 3 цветов. Двумя цветами он не может быть окрашен в принципе.

Используя 4 цвета, получаем  $24 + 4 * 12 = 72$  вариантов раскраски: при использовании всех 4 цветов, есть  $4! = 24$  корректных способа (каждое присвоение 4 цветов для любого графа из 4 вершин является корректным); и для каждой 3 цветов из этих 4 есть 12 способов раскраски.

## 2. Алгоритмы раскраски графа

Алгоритм раскраски графа позволяет находить (точное или приближенное) значение хроматического числа произвольного графа и соответствующую этому значению раскраску вершин.

Граф  $G$  называют  *$r$ -хроматическим*, если его вершины могут быть раскрашены с использованием  $r$  цветов (красок) так, что не найдется двух смежных вершин одного цвета.

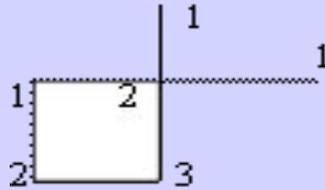
Наименьшее число  $r$ , такое, что граф  $G$  является  $r$ -хроматическим, называется *хроматическим числом графа  $G$* .

*Задача нахождения хроматического числа графа называется задачей о раскраске (или задачей раскраски) графа.*

Соответствующая этому числу раскраска вершин разбивает множество вершин графа на  $r$  подмножеств, каждое из которых содержит вершины одного цвета.

Эти множества являются независимыми, поскольку в пределах одного множества нет двух смежных вершин.

Правильная раскраска графа  $G$  может выглядеть следующим образом:



Рассмотрим **последовательный метод**, основанный на упорядочивании множества вершин.

### Формальное описание

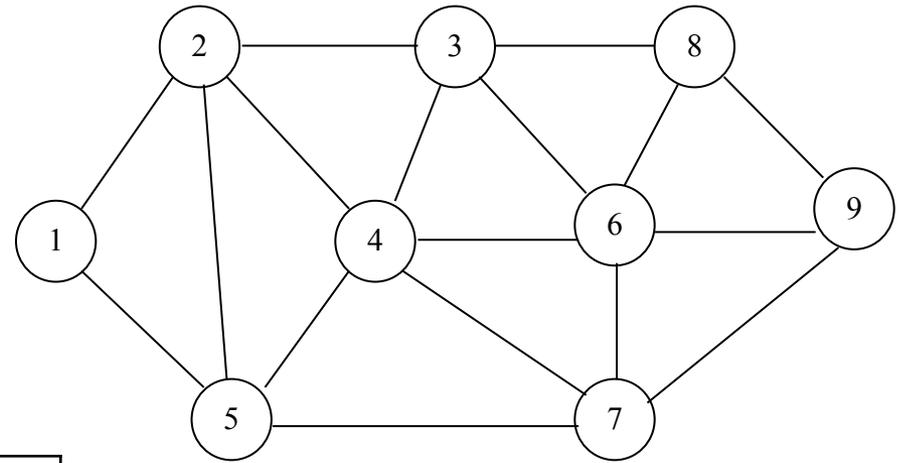
1. Упорядочить вершины по невозрастанию степени.
2. Окрасить первую вершину в цвет 1.
3. Выбрать цвет окраски 1.
4. Пока не окрашены все вершины, повторять п.4.1.-4.2.:
  - 4.1. Окрасить в выбранный цвет всякую вершину, которая не смежна с другой, уже окрашенной в этот цвет.
  - 4.2. Выбрать следующий цвет.

# *Раскраска графа по степеням вершин*

## *Алгоритм*

1. Упорядочить вершины по степеням начиная с наибольшей степени вершины.
2. Задать начальное значение счетчика  $k=1$ .
3. Первую вершину окрашиваем в цвет  $\lambda_k$  и заносим в букет  $B(k)$ .
4. Просматриваем следующую неокрашенную вершину, если она несмежная с вершинами букета  $B(k)$  то окрашиваем ее в цвет  $\lambda_k$ , иначе пропускаем.
5. Проверяем количество просмотренных вершин, если  $i \leq n$  ( $i$ -номер текущей вершины), то возврат в п.4, иначе  $k=k+1$  и просмотр списка начинается заново исключая окрашенные вершины.
6. Проверка на окончание поиска: если неокрашенных вершин не осталось, то конец поиска, иначе п.5.

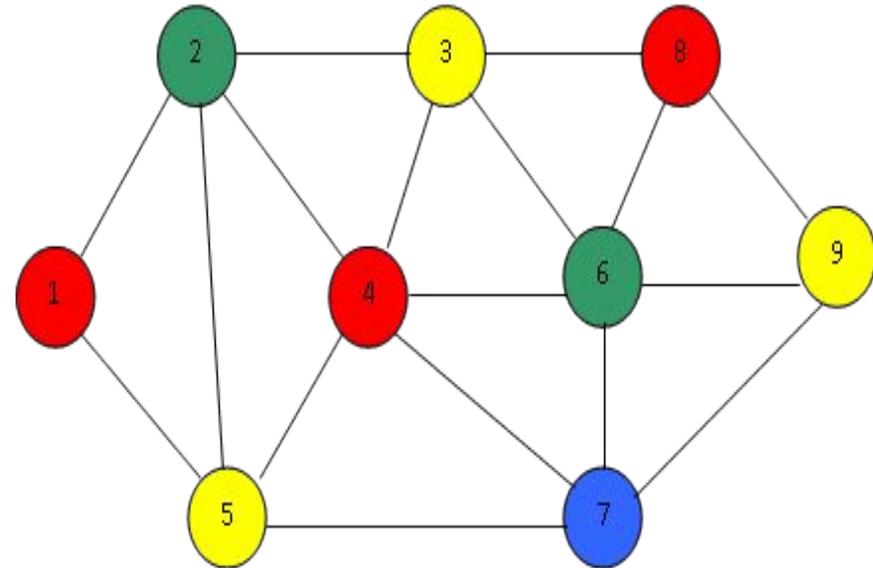
*Пример: задан граф*



*Раскрасить по степеням вершин*

*Составим таблицу:*

Вершина $v_i$	Степень $\Delta$	$B(1)$	$B(2)$	$B(3)$	$B(4)$
4	5	4			
6	5		6		
3	4			3	
2	4		2		
5	4			5	
7	4				7
8	3	8			
9	3			9	
1	2	1			
Цвет $\lambda$		1	2	3	4



В этом простейшем из методов вершины вначале располагаются в порядке невозрастания их степеней.

Первая вершина окрашивается в цвет 1; затем список вершин просматривается сверху вниз (по невозрастанию степеней) и в цвет 1 окрашивается всякая вершина, которая не смежна с другой, уже окрашенной в этот цвет.

Потом возвращаемся к первой в списке неокрашенной вершине, окрашиваем ее в цвет 2 и снова просматриваем список вершин сверху вниз, окрашивая в цвет 2 любую неокрашенную вершину, которая не соединена ребром с другой, уже окрашенной в цвет 2 вершиной. Аналогично действуем с цветами 3, 4 и т. д., пока не будут окрашены все вершины.

Число использованных цветов будет тогда приближенным значением хроматического числа графа.

*Приведенный алгоритм можно модифицировать.*

Для этого после каждого шага нужно упорядочивать неокрашенные вершины.

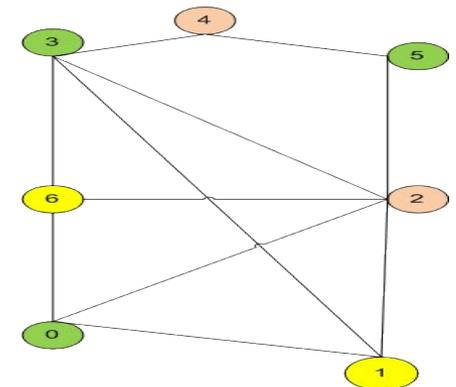
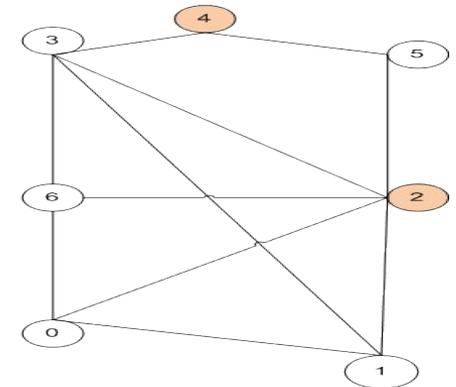
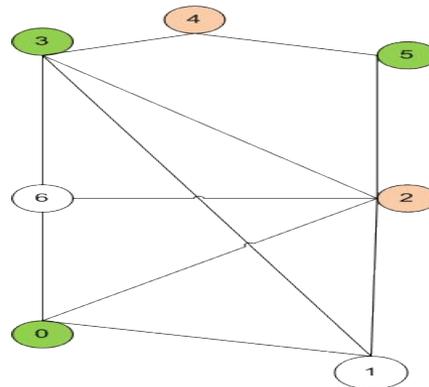
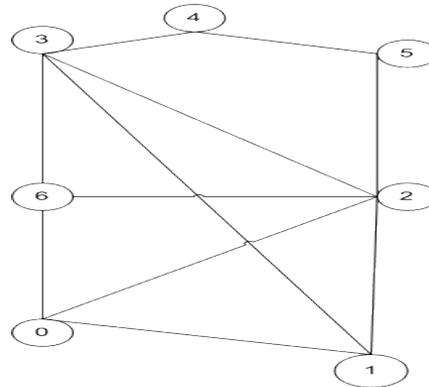
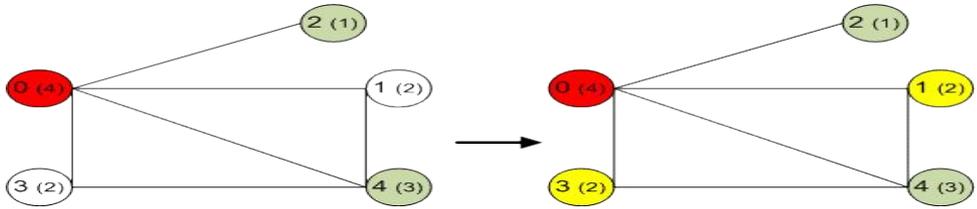
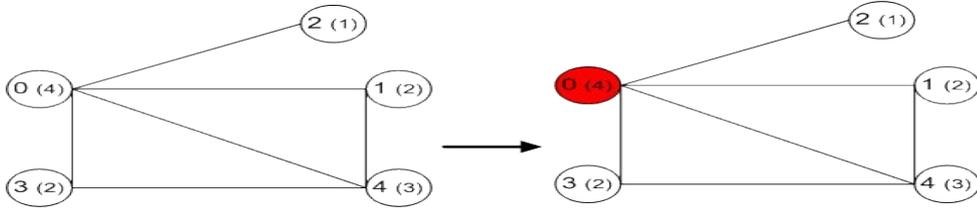
Простая модификация описанной процедуры состоит в переупорядочивании неокрашенных вершин по невозрастанию их относительных степеней.

Под относительными степенями понимаются степени соответствующих вершин в неокрашенном подграфе исходного графа.

В данной модификации предполагалось, что если две вершины имеют одинаковые степени, то порядок таких вершин случаен. Такие вершины можно также упорядочить, но уже по двухшаговым степеням.

Двухшаговую степень определим как сумму относительных степеней инцидентных вершин. Аналогично можно поступать и далее.

# Примеры раскрасок графов



# *Алгоритм с использованием независимого множества вершин*

*Независимое множество вершин  $S$  – это такое подмножество вершин, в котором никакие две вершины этого подмножества не соединены ребром.*

$$S = \{v_1, v_2, \dots, v_n\} \wedge \forall v_i \cap M(v_i) = \emptyset$$

*Максимальное независимое множество вершин  $S_{max}$  - это независимое множество вершин, не являющееся собственным подмножеством никакого другого независимого множества вершин. Оно содержит наибольшее число вершин. В графе может быть несколько  $S_{max}$ .*

*Для отыскания  $S_{max}$  существует несколько методов. Точный метод основан на булевой алгебре. Он достаточно сложен.*

## *Рассмотрим приближенный метод*

*Метод выбора  $S_{max}$  из нескольких  $S$ : от каждой вершины графа определяются все независимые множества вершин, из которых выбирается множество с максимальной мощностью.*

## *Алгоритм определения независимого множества $S$*

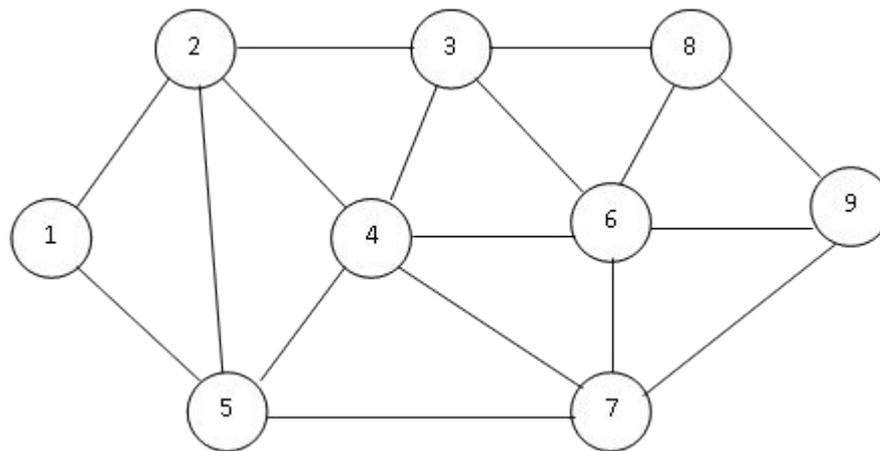
1. Выбирается начальная вершина  $v_0$  и заносится в  $S$ .
2. Рассматривается следующая вершина  $v_i$ , если  $v_i$  несмежная с  $S$ , то она включается в  $S$ , иначе не включается.
3. Проверка на окончание: если просмотрены все вершины, то — конец, иначе п.2.

## *Алгоритм раскраски с использованием независимого множества вершин*

Рассмотрим следующую схему рекурсивной процедуры  $P$ :

1. Выбрать в графе  $G$  некоторое максимальное независимое множество вершин  $S_{max}$ .
2. Окрасить вершины множества  $S_{max}$  в очередной цвет.
3. Применить процедуру  $P$  к графу  $G \setminus S_{max}$ .

*Пример: задан граф  
Раскрасить с использованием  
независимых множеств вершин*



*Определим  $S_{max}$*

$$S_{max1} = \{1, 3, 7\} - 1 \text{ цвет}$$

$$S_{max2} = \{4, 9\} - 2 \text{ цвет}$$

$$S_{max3} = \{5, 8\} - 3 \text{ цвет}$$

$$S_{max4} = \{2, 6\} - 4 \text{ цвет}$$

