ЛЕКЦИЯ 11

План лекции 1.
Задачи раскраски
2.Основные определения
3. Хроматическое число
4. Хроматическое число и стандартные характеристики
5. Хроматическое число и плотность графа
6. Хроматическое число и число независимости графа
7. Третья нижняя оценка хроматического числа
8. Алгоритм последовательной раскраски
9. Алгоритм последовательной раскраски с упорядочиванием множества вершин
10. Модификация алгоритма последовательной раскраски
11. Рекурсивная процедура последовательной раскраски
12. «Жадный» алгоритм раскраски
13. Результаты работы алгоритмов последовательной раскраски
14. Теорема Брукса
15. Теоремы о шести, пяти и четырех красках
16. Задача распределения оборудования
17. Задача составления расписания

Задачи раскраски

Задачи раскраски вершин или ребер графа занимают важное место в теории графов.

К задаче построения раскраски графа сводится целый ряд практических задач.

Одна из областей – составление расписаний.

- расписания для **образовательных учреждений;**
- расписания в спорте;
- планирование встреч, собраний, интервью;
- расписания транспорта, в том числе — авиатранспорта;
- расписания для коммунальных служб;
- и прочие.

Правильная раскраска

Пусть $G=(V,E)$ — конечный граф, а k — некоторое натуральное число.

Вершинная раскраска. Произвольная функция вида $f:V\to N_k$, где $N_k=\{1,2,...,k\}$, называется *вершинной* k*-раскраской*, или просто k*-раскраской* графа G.

Правильная раскраска. Раскраска называется *правильной*, если цвета смежных вершин не совпадают, т. е. для всех $(u, v) \in E$ справедливо $f(u) \neq f(v)$.

Раскрашенный граф. Граф, для которого существует правильная k-раскраска, называется *раскрашенным графом.*

Базовый принцип оптимизации раскраски

Если функция f **не взаимно однозначная**, то при $|V| = k$ фактически может быть использовано **меньше** чем k цветов.

Правильная раскраска – это разбиение множества вершин

Правильную k-раскраску можно рассматривать как разбиение множества вершин V графа G на классы $V_1 \cup V_2 \cup \ldots \cup V_l = V$, где $l \leq k$, $V_i \neq \varnothing$, $i = 1, 2, \ldots, l$.

Каждый класс V_i — независимое множество, а сами классы называются *цветными классами.*

Хроматическое число

Определение. Минимальное число k, при котором существует правильная k-раскраска графа G, называется *хроматическим числом* этого графа и обозначается $X_p(G)$.

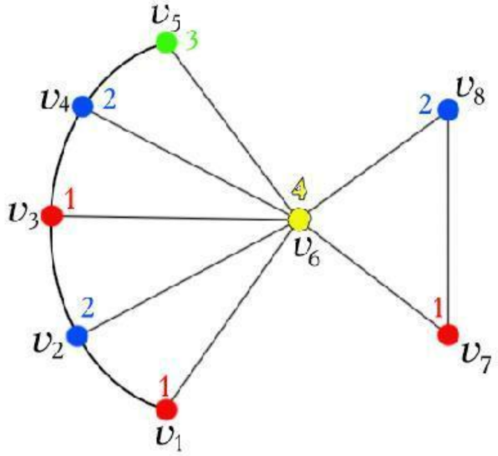
Определение. Если $X_p(G) = k$, то граф G называется k*-хроматическим.* То есть его вершины можно раскрасить k разными цветами так, что у любого ребра концы будут разного цвета.

Определение. Правильная k-раскраска графа G при $k = X_p(G)$ называется *минимальной.*

Определение. Хроматическое число несвязного графа равно максимальному из хроматических чисел его компонент связности.

Пример.

Рассмотрим граф G, изображенный на рисунке, где показана одна из правильных k-раскрасок. Натуральными числами 1,2,3,4 обозначены краски соответствующих вершин.

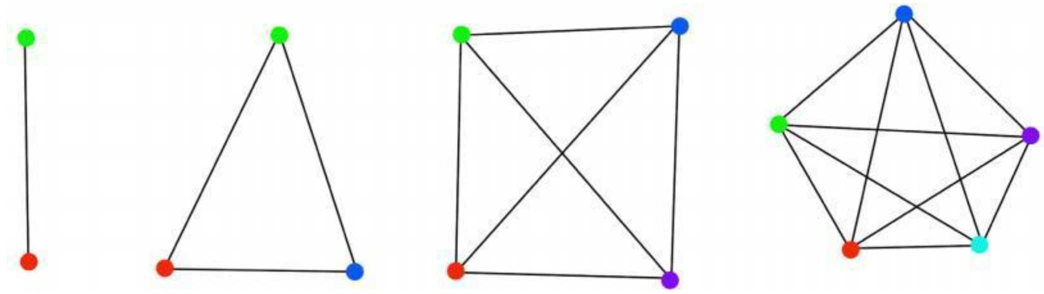


Хроматические числа некоторых графов

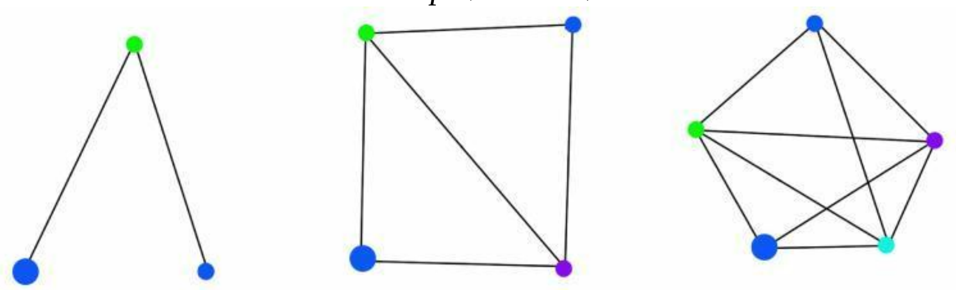
Для некоторых простых графов нетрудно найти хроматические числа.

Примеры

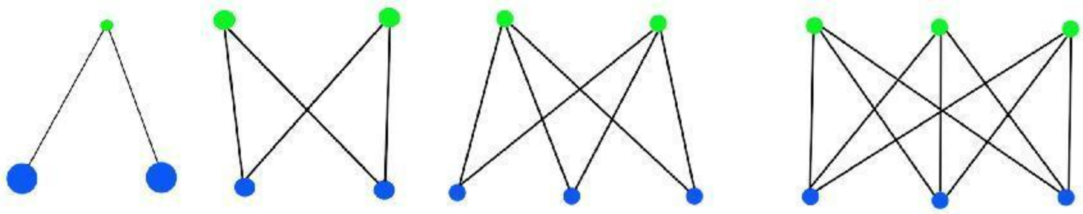
1. Полный граф K_n, состоящий из n вершин, имеет хроматическое число $X_p(K_n) = n$



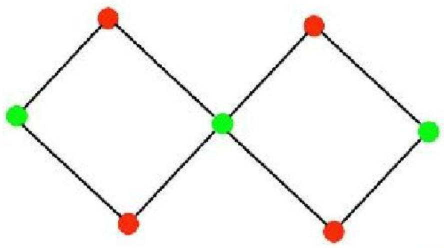
2. Полный граф $K_n - e$, состоящий из n вершин с одним недостающим ребром, имеет хроматическое число $X_p(K_n - e) = n - 1$



3. Полные двудольные графы $K_{m,n}$, состоящие из долей $|A| = m$ и $|B| = n$, имеют хроматическое число $X_p(K_{m,n}) = 2$



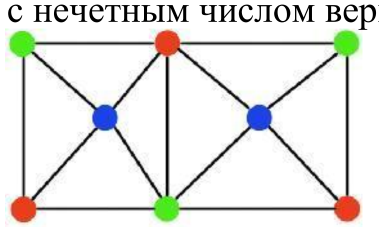
Теорема. Непустой граф является бихроматическим тогда и только тогда, когда он не имеет циклов нечетной длины.



1-хроматический граф – пустой граф. 

2-хроматический граф — двудольный непустой граф, 2-хроматические графы, как правило, называют *бихроматическими.*

3-хроматический граф — при каких условиях граф является 3-хроматическим, неизвестно, хотя легко найти примеры таких графов. К ним относятся циклические графы с нечетным числом вершин.



Когда граф имеет n вершин, то его хроматическое число не превышает n

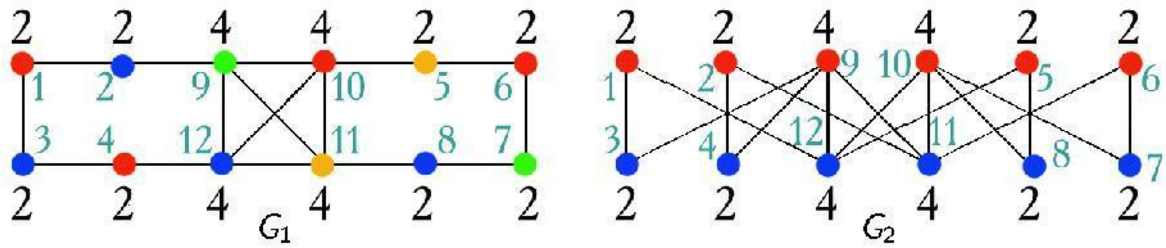
Когда граф имеет подграф K_m , то его хроматическое число не меньше m .

Значительно больше информации о хроматическом числе графа можно получить, если известны степени каждой его вершины.

Обозначим через $r = \max\limits_{v \in V} \left(\deg(v)\right)$ максимальную степень вершины графа $G = (V, E)$.

Хроматическое число и стандартные характеристики

В общем случае хроматическое число графа нельзя вычислить, зная только его стандартные числовые характеристики: *число вершин, ребер, компонент связности, распределение степеней вершин.*



Рассмотрим графы G_1 и G_2 . Каждый из них имеет 12 вершин, в том числе 4 вершины степени 4 и восемь вершин степени 2, шестнадцать ребер, одну компоненту связности. Но, как легко понять, $x(G_1) = 4$, а $x(G_2) = 2$, поскольку G_1 содержит в качестве подграфа граф K_4 .

Так как граф G_2 — двудольный, имеем $x(G_2) = 2$.

Поэтому в дальнейшем речь пойдет об оценках, а не о точных значениях хроматического числа.

Хроматическое число и плотность графа Под нижними оценками хроматического числа будем понимать неравенства вида $X(G) \geq c$, где c — некоторая константа, вычисляемая по графу G .

Верхняя оценка хроматического числа — это неравенства вида $X(G) \leq c$, где c имеет тот же смысл.

Определение. Максимальное число вершин, порождающих полный подграф в графе G, называется ***плотностью*** G и обозначается через $\omega(G)$.

Первая нижняя оценка

Для произвольного графа G справедливо неравенство $X(G) \geq \omega(G)$.

Хроматическое число и число независимости графа

Определение. Любое множество попарно несмежных вершин графа G называется *независимым множеством.*

Определение. Максимальное число вершин в независимом множестве называется *числом независимости* графа G и обозначается через $\beta(G)$.

Число независимости графа — это понятие, противоположное по смыслу понятию плотности графа. Если G — обыкновенный граф, а $\overline{G}$ — его дополнение, то $\beta(G) = \omega(\overline{G})$.

Вторая нижняя оценка

Для произвольного графа G выполнено неравенство

$$X(G) \geq \frac{n(G)}{\beta(G)}$$

Третья нижняя оценка хроматического числа

Существуют **нижние оценки** хроматического числа графа, использующие только легко вычисляемые его характеристики. Приведем без доказательства одну из них.

Третья нижняя оценка

Если G – обыкновенный граф и

$n = n(G)$ – количество вершин графа G,

$m = m(G)$– количество ребер графа G,

то хроматическое число $X(G) \geq \dfrac{n^2}{n^2 - 2m}$

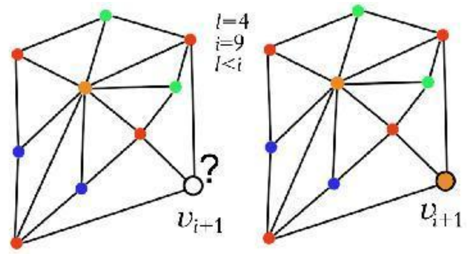
Легко понять, что в полном графе (а значит, и в любом обыкновенном графе) удвоенное число ребер меньше квадрата числа вершин, и потому число, стоящее в знаменателе в правой части неравенства, всегда положительно.

Как видно из описанных выше результатов, **задачи определения хроматического числа** графа и построения минимальной раскраски произвольного графа **достаточно сложны**, а эффективные алгоритмы их решения неизвестны. Рассмотрим простой алгоритм построения правильной раскраски, который в некоторых случаях дает раскраски, близкие к минимальным.

Алгоритм последовательной раскраски

1. Если вершины v_1, v_2,...,v_i раскрашены l цветами 1,2,...,l; $l \leq i$, то новой произвольно взятой вершине v_{i+1} припишем минимальный цвет, не использованный при раскраске смежных с ней вершин.

Раскраска, к которой приводит описанный алгоритм, называется *последовательной*.



Алгоритм последовательной раскраски с упорядочиванием множества вершин

1. Упорядочить вершины по невозрастанию степени.

2. Выбрать цвет окраски 1.

3. Окрасить первую вершину в цвет 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 вершиной, уже окрашенной в этот цвет.

4.2. Выбрать следующий цвет.

5. Возвратиться к первой в списке неокрашенной вершине.

Число использованных цветов будет тогда приближенным значением хроматического числа графа.

Модификация алгоритма последовательной раскраски

Определение. Относительная степень – это степень неокрашенных вершин в неокрашенном подграфе исходного графа.

Определение. Двухшаговая относительная степень – сумма относительных степеней смежных вершин в неокрашенном подграфе**.**

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

В данной модификации предполагалось, что если две вершины имеют одинаковые степени, то порядок таких вершин случаен. Их можно упорядочить по двухшаговым степеням. Двухшаговую степень определим как сумму относительных степеней инцидентных вершин. Аналогично можно поступать и далее.

Раскраска графа методом А.П. Ершова

Андрей Петрович Ершов (1931 -1988 гг.), крупнейший ученый в области теоретического программирования, внес большой вклад в развитие информатики в нашей стране. Им создан алгоритм раскраски графа, выделяющийся оригинальной эвристической идеей.

Введем ряд определений.

Для данной вершины $v \in V$ графа $G(V, E)$ назовем все смежные с ней вершины окрестностью 1-го порядка — $R_1(v)$.

Все вершины, находящиеся от v на расстоянии два, назовем окрестностью 2-го порядка — $R_2(v)$.

Граф $G(V, E)$, у которого для вершины $v \in V$ все остальные вершины принадлежат окрестности $R_1(v)$ назовем граф-звездой относительно вершины v .

Идея алгоритма

Окрашивание в краску α вершины v образует вокруг нее в $R_1(v)$ «мертвую зону» для краски α . Очевидно, при минимальной раскраске на каждую краску должно приходиться наибольшее число вершин графа. Для этого необходимо, чтобы мертвые зоны хотя бы частично перекрывались между собой. Перекрытие мертвых зон двух несмежных вершин v_1 и v_2 достигается только тогда, когда одна из них находится в окрестности $R_2(v_1)$ от другой, $v_2 \in R_2(v_1)$.

Таким образом, суть алгоритма состоит в том, чтобы на очередном шаге выбрать для раскраски краской α вершину $v_2 \in R_2(v_1)$. Этот процесс повторять до тех пор, пока краской α не будут покрашены все возможные вершины графа.

Графически окрашивание вершин v_1 и v_2 одной краской можно отобразить как «склеивание» этих вершин.

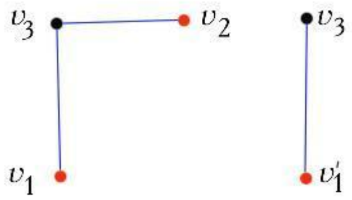


Рис. 14.1. Пример объединения двух вершин: $v_1' := v_1 \cup v_2$

При этом уменьшается на единицу число вершин в графе G и уменьшается число ребер.

Алгоритм

1.Положить $i := 0$.

2.Выбрать в графе G произвольную неокрашенную вершину v .

3.Положить $i := i + 1$.

4.Окрасить вершину v в краску i .

5.Окрашивать в краску i неокрашенные вершины графа G , выбирая их из $R_2(v)$, пока граф не превратится в граф-звезду относительно v .

6.Проверить, остались ли неокрашенные вершины в графе G . Если да, то перейти к п. 2, иначе - к п. 7.

7.Получен полный граф K_i . Хроматическое число графа $X(K_i) = i$. Конец алгоритма.

Как следует из алгоритма, после того, как в первую выбранную вершину стянуты все возможные и граф превратился в граф-звезду относительно этой вершины, выбирается произвольным образом вторая вершина и процесс повторяется до тех пор, пока граф не превратится в полный.

Полный граф - это граф-звезда относительно любой вершины. Хроматическое число полного графа равно числу его вершин.

Пример раскраски методом А.П. Ершова

На рис.14.2 изображен граф G , на примере которого будем рассматривать работу эвристического алгоритма Ершова.

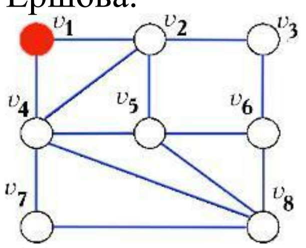


Рис.14.2. Исходный граф G

Выберем произвольную вершину, например, v_1 . Окрестность 2-го порядка $R_2(v_1) = \{v_3, v_5, v_7, v_8\}$. Склеим вершину v_1 например, с вершиной v_3 : $v_1' = v_1 \cup v_3$. Получим граф G_1 изображенный на рис. 14.3.

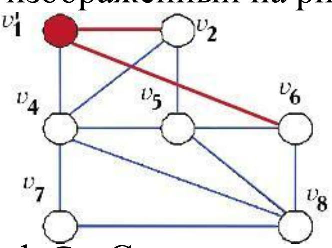
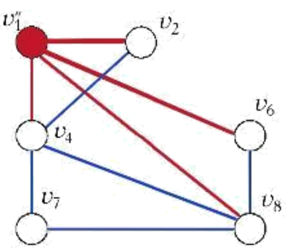


Рис.14.3. Граф G_1. Склеены вершины v_1 и v_3

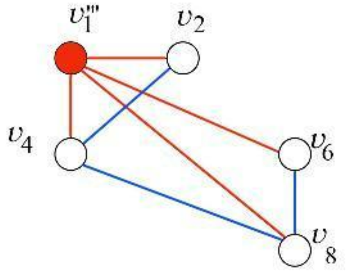
Рассмотрим теперь граф G_1. Окрестность второго порядка вершины v_1' определяется множеством $R_2(v_1') = \{v_5 .v_7 ,v_8\}$. Склеим вершину v_1', например, с вершиной v_5 : $v_1'' := v_1' \cup v_5$. Получим графG_2 , изображенный на рис. 14.4.



$'$

Рис.14.4. Граф G_2 . Склеены вершины v_1 и v_5

В графе G_2 определим окрестность второго порядка для вершины v_1'': $R_2(v_1'') = \{v_7\}$. Склеим вершину v_1'' с вершиной v_7 : $v_1''' = v_1'' \cup v_7$. Получим граф G_3 , изображенный на рис. 14.5, являющийся граф-звездой относительно вершины v_1'''.



$''$

Рис.14.5. Граф G_3 . Склеены вершины v_1 и v_7

В графе G_3 выберем, например, вершину v_2 для окрашивания второй краской. Окрестность 2-го порядка $R_2(v_2) = \{v_6, v_8\}$. Склеим вершину v_2 , например, с вершиной v_6 : $v_1''' = v_1'' \cup v_6$. Получим граф G_4 , изображенный на рис.14.6.

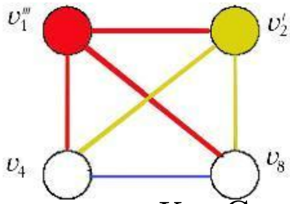


Рис.14.6. Граф G_4 эквивалентен K_4 . Склеены вершины v_1'' и v_6

Граф G_4 является полным четырехвершинным графом K_4. Следовательно, граф G_4 на рис.14.6 раскрашивается в четыре цвета. В первый (красный) цвет раскрашивается вершина v_1 и склеенные с ней вершины: v_3 ,v_5 и v_7 . Во второй (желтый) цвет раскрашивается вершина v_2 и склеенная с ней вершина v_6 . В третий (зеленый) цвет раскрашивается вершина v_4 и в четвертый (белый) цвет раскрашивается вершина v_8

В итоге получаем правильно раскрашенный граф, показанный на рис.14.7.

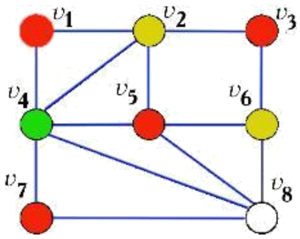


Рис.14.7.Граф G , раскрашенный с помощью алгоритма раскраски А.П.Ершова

Рекурсивная процедура последовательной раскраски

1. Фиксируем порядок обхода вершин.
2. Идем по вершинам, используя такой наименьший цвет, который не вызовет конфликтов.
3. Если уже использованный цвет выбрать не получается, то только тогда вводим новый цвет.

В процедуре используется рекурсивный вызов процедуры окраски следующей вершины в случае успешной окраски предыдущей вершины.

```
procedure visit(i:Byte);
 Var i,Cmax:Byte;
 Function NiceColor:Boolean;
 Var CN:Boolean;
 Begin
  CN:=true;
  For j=1 to n do
  If (A[j,i]=1) AND (color[j]=c) then CN:=false;
 End;
begin
 if i = n + 1 then Print
 else begin
  If color[i]=0 then
  begin
   for c:=color[i]+1 to Cmax
   do if NiceColor then
   begin
    color[i]:=c;
    visit(i+1);
   end;
  end;

end;

end;
```

«Жадный» алгоритм раскраски

Пусть дан связный граф $G(V, E)$.

1. Зададим множество *monochrom* $:= \varnothing$, куда будем записывать все вершины, которые можно окрасить одним цветом.

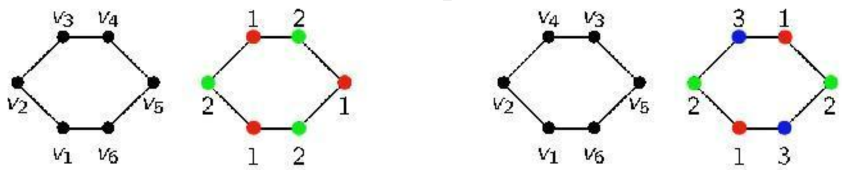
2. Просматриваем все вершины и выполняем следующий «жадный» алгоритм

Procedure Greedy
For (для каждой незакрашенной вершины $v \in V$) **do**
If *v не смежна с вершинами из monochrom* **then**
begin
  $color(v) :=$ цвет*;*
 $monochrom := monochrom \cup \{v\}$
end

Результаты работы алгоритмов последовательной раскраски

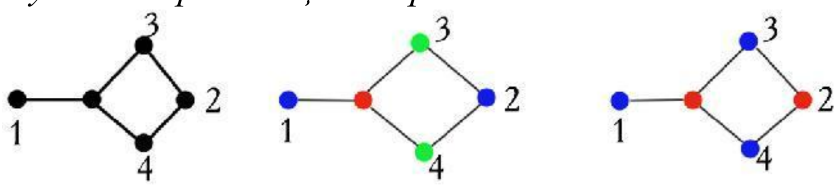
Полученная раскраска всегда правильна, но не всегда оптимальна даже для простых графов.
Она существенно зависит от того, в каком порядке выбираются вершины.

На первом рисунке получился оптимальный результат (2 краски), а на втором рисунке использовано больше красок.

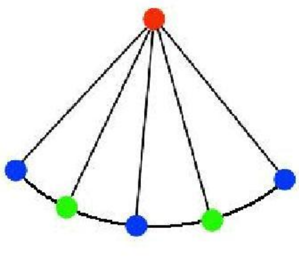


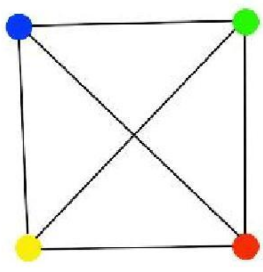
Закрасив вершину 1 в синий цвет, а затем, пропустив вершину 2, закрасить в синий цвет вершины 3 и 4, можно получить 2 краски. Но "жадный" алгоритм, основываясь на порядковой очередности вершин, закрасит в красный цвет вершины 1 и 2, для раскраски графа теперь нужно 4 краски.

Поэтому актуальна верхняя оценка хроматического числа

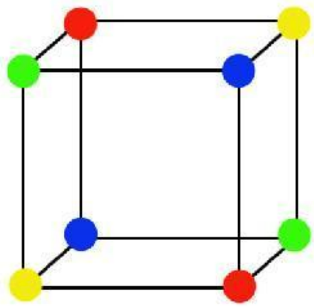


Теорема 2. Для всякого графа G имеет место неравенство $X_p(G) \le r + 1$, *где* $r = \max\limits_{v \in V}(\deg(v))$ максимальная степень вершины графа $G = (V, E)$.





Следствие. Всякий кубический граф раскрашивается с помощью четырех красок.

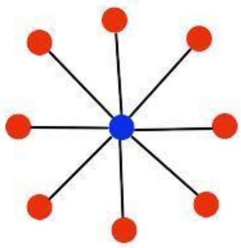


Более общий результат, чем рассмотренная теорема, дает такая теорема.

Теорема Брукса. *Если G — связный неполный граф и* $r \geq 3$, *то* $X_p(G) \leq r$.

Хотя обе теоремы и дают определенную информацию о хроматическом числе графа, но их оценки довольно неточны.

Действительно, **звездный граф** K_{1n}, который согласно теореме Брукса **раскрашивается** n **красками**, на самом деле является **бихроматичным**.



Эта ситуация сильно упрощается, если ограничиться ***планарными графами*** (планарный граф – изоморфный плоскому (плоский – в котором ребра пересекаются лишь в вершинах)). В этом случае легко доказать такой достаточно общий и сильный факт.

Теоремы о шести, пяти и четырех красках

Теорема о шести красках. Для всякого планарного графа G имеет место неравенство $X_p(G) \leq 6$.

Более детальный анализ путей снижения верхней границы хроматического числа приводит к так называемой *теореме о пяти красках.*

Теорема о пяти красках. Для всякого планарного графа G имеет место неравенство $X_p(G) \leq 5$.

Теорема о четырех красках. Каждый планарный граф без петель и кратных ребер является не более чем 4-хроматическим.

Проблема четырех красок оставалась нерешенной в течение многих лет. Утверждается, что эта теорема была доказана с помощью хитроумных рассуждений и компьютерной программы в 1976 году (**Kenneth Appel and Wolfgang Haken. Every Planar Map is Four Colorable. Contemporary Mathematics 98, American Mathematical Society, 1980**).

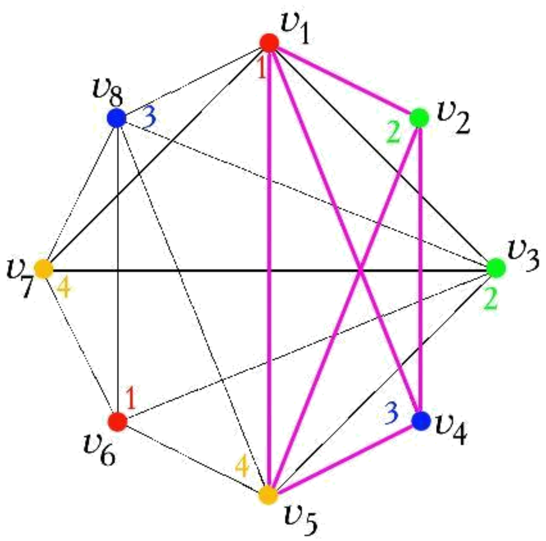
ЗАДАЧА РАСПРЕДЕЛЕНИЯ ОБОРУДОВАНИЯ

На предприятии планируется выполнить 8 работ v_1 , v_2 ,...,v_8 . Для выполнения этих работ необходимы механизмы a_1 , a_2 ,...,a_6 . Использование механизмов для каждой из работ определяется следующей таблицей:

Меха-	Работа							
низм	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
a_1	+		+				+	+
a_2		+		+				
a_3			+			+	+	
a_4	+	+		+	+			
a_5			+		+			+
a_6					+	+		+

Ни один из механизмов не может быть использован одновременно на двух работах. Выполнение каждой работы занимает 1 час. Как распределить механизмы, чтобы суммарное время выполнения всех работ было минимальным и каково это время?

Решение. Рассмотрим граф G , вершинами которого являются планируемые работы v_1 , v_2 ,...,v_8 , а ребра соединяют работы, в которых участвует хотя бы один общий механизм (и которые, по этой причине, нельзя проводить одновременно).



Вершины v_1 , v_2 , v_4 ,v_5 порождают подграф графа G , изоморфный K_4 .

Следовательно, $X\left\{G\right\} \geq 4$. Следовательно, $X\left(G\right)$.
Таким образом, все работы можно выполнить за 4 часа.
Для этого, в соответствии с найденной раскраской графа G, надо в 1-й час выполнять работы v_1 и v_6 ,

во 2-й час — работы v_2 и v_3 ,

в 3-й час — работы v_4 и v_8 ,

в 4-й час — работы v_5 и v_7 .

Механизм	Работа							
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
a_1	+		+				+	+
a_2		+		+				
a_3			+			+	+	
a_4	+	+		+	+			
a_5			+		+			+
a_6					+	+		+

Задача составления расписания

Условие задачи

Нужно прочесть лекции по четырем предметам двум группам студентов.
Некоторые из лекций не могут читаться одновременно.
Существует ряд причин, по которым это невозможно:

1. Лекцию по данному предмету в разных группах читает один и тот же лектор.
2. Две лекции одной и той же группе одновременно читаться не могут.
3. Разные лекции должны проходить в одном и том же помещении.

Требуется составить расписание так, чтобы чтение всех лекций заняло минимально возможное время (в качестве «единицы времени» в данной задаче естественно рассматривать одну пару).

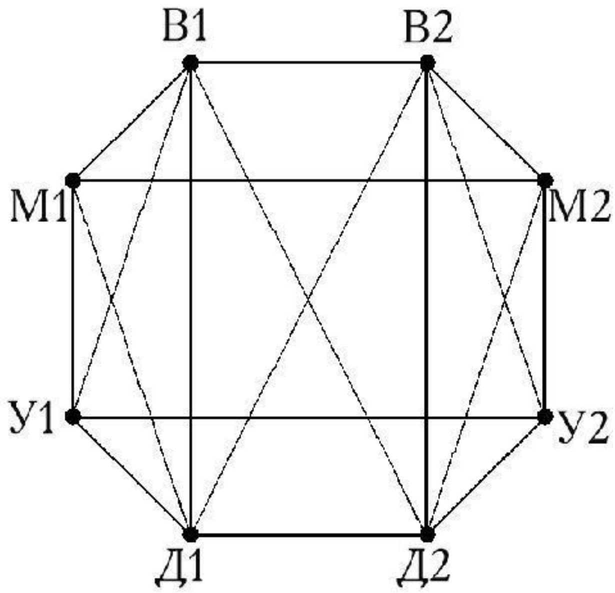
Конкретизируем постановку задачи

1. Будем рассматривать две группы студентов: 1 и 2.

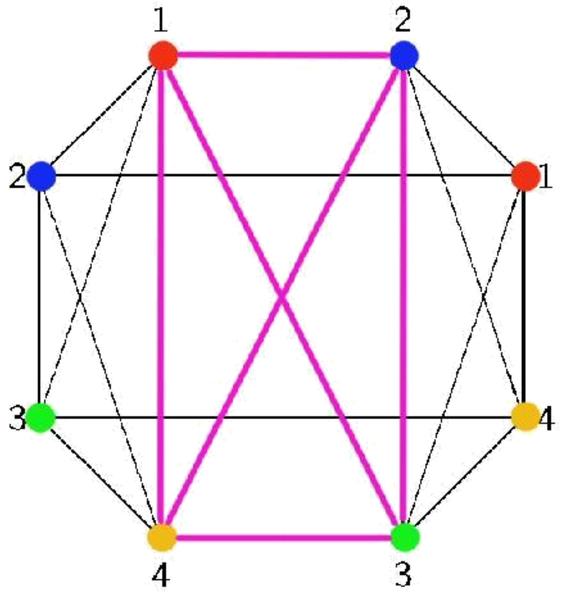
2. Читаемые предметы:
- вычислительные методы – читает преп. X,
- дискретная математика – читает преп. X,
- математический анализ – читает преп. Y,
- украинский язык – читает преп. Z.

Найти минимальное число пар, в которые можно «уложить» все занятия, и составить соответствующее расписание

Решение. Построим граф с вершинами В1, В2, Д1, Д2, М1, М2, У1 и У2 (буква соответствует предмету, а цифра — номеру группы).
Соединим ребрами вершины, соответствующие парам, которые нельзя проводить одновременно.
Получим граф, показанный на рисунке:



Вершины В1, В2, Д1 и Д2 этого графа порождают в нем подграф, изоморфный графу K_4 . Следовательно, хроматическое число нашего графа не меньше 4. На рисунке указана правильная раскраска нашего графа в 4 краски.



Следовательно, хроматическое число графа равно 4, т. е. все занятия можно провести за 4 пары. Соответствующее расписание указано в таблице.

	Группа 1	Группа 2
1 пара	Выч. методы	Матем. анализ
2 пара	Матем. анализ	Выч. методы
3 пара	Украинский яз.	Дискр. матем
4 пара	Дискр. матем.	Украинский яз.