Раскраска вершин графа

. ***Хроматическое число***

Пусть $G=(V,E)$ – неориентированный граф. Произвольная функция $f:V\to\{1,2,...,k\}$, где k принадлежит множеству натуральных чисел, называется вершинной k-раскраской графа G. Раскраска называется правильной, если $f(u)\neq f(v)$, для любых смежных вершин u и v. Граф, для которого существует правильная k-раскраска, называется k-раскрашиваемым. Минимальное число k, при котором граф G является k-раскрашиваемым, называется *хроматическим числом* графа и обозначается $\chi(G)$. Очевидно, что граф G имеет n-раскраску ($|V| = n$) и для любого p из интервала $\chi(G)\leq p\leq n$ существует p-раскраска графа G.

Через $\Delta(G)$ обозначим наибольшую из степеней вершин графа G.

Теорема. Для любого графа G верно неравенство $\chi(G)\leq 1+\Delta(G)$.

Пример. На рис. 11.1 приведен граф, его хроматическое число $\chi(G)$ равно трем. Меньшим количеством цветов граф правильно раскрасить нельзя из-за наличия треугольников.

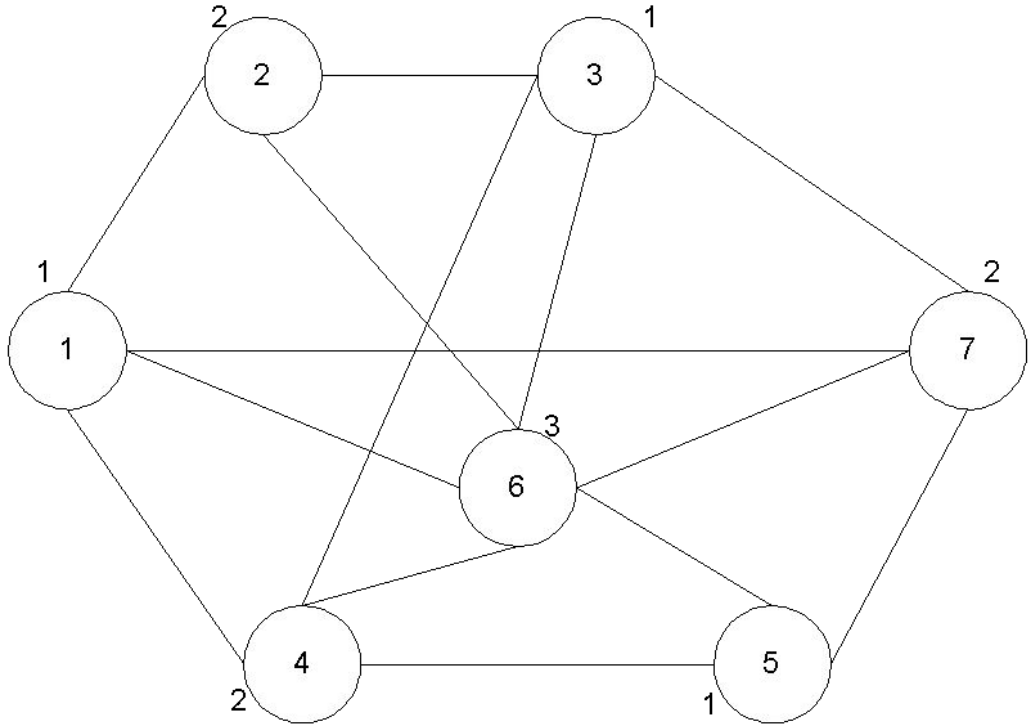

Рис. 11.1. Пример графа для иллюстрации понятия «хроматическое число». Рядом с «кружками» - вершинами графа - указаны номера цветов

Мы говорим о раскраске вершин графа, однако исторически проблема связана с раскраской карт. В 1879 г. А. Кэли в работе, посвященной проблеме раскраски карт, сформулировал следующую гипотезу.

Гипотеза четырех красок: всякая карта 4-раскрашиваема.

Уточним понятие «раскрашиваемая карта». С точки зрения теории графов это плоский граф, грани которого раскрашены так, что любым двум смежным граням присвоены разные цвета. Таким образом, грани, имеющие общее ребро, окрашиваются в разные цвета, а грани, имеющие только общую вершину или не имеющие общих вершин, могут окрашиваться в один цвет. Очевидно и то, что граф должен быть без мостов (ребра, удаление которых приводит к увеличению количества компонент связности). По любой карте строится граф с вершинами, соответствующими раскрашиваемым областям карты и с очевидным отношением смежности. Таким образом, проблема раскраски карты сводится к задаче раскраски вершин графа.

В 1880 году А. Кемпе привел доказательство гипотезы. Только в 1890 году Р. Хивуд нашел ошибку в рассуждениях и доказал теорему.

Теорема. Каждый планарный граф 5-раскрашиваем.

В 1969 году проблема четырех красок Х. Хеешем была сведена к определению раскрашиваемости большого, но конечного множества U «неустранимых конфигураций» [11, с. 263]. Он показал, что в любом максимальном плоском графе G найдется подграф $\widetilde{G}$, изоморфный некоторой конфигурации из U и такой, что если $\widetilde{G}$ 4-раскрашиваемым, то и G также 4-раскрашиваемый. Постепенно количество таких «неустранимых конфигураций» свели к 1482. В 1976 году коллективу математиков и программистов удалось найти четырехцветные раскраски для всех графов из U (потребовалось порядка 2000 часов работы мощной ЭВМ). Является ли их утверждение о доказанности гипотезы – трудно сказать. Ф. Харари пишет: «Гипотезу четырех красок можно с полным основанием назвать еще «болезнью четырех красок», так как она во многом похожа на заболевание. Она в высшей степени заразна. Иногда она протекает сравнительно легко, но в некоторых случаях приобретает затяжной или даже угрожающий характер…» [27, с. 151].

Метод правильной раскраски

Его суть достаточно проста и сводится к одной фразе – «закрашиваем очередную вершину в минимально возможный цвет». Введем следующие структуры данных.

Const Nmax=100; {Максимальное количество вершин графа.}

Type V=0..Nmax;

Ss=Set Of V;

MyArray=Array[1..Nmax] Of V;

Var Gr:MyArray;{*Gr* – для каждой вершины графа определяется номер цвета.}

Для примера на рис. 11.1 массив *Gr* имеет вид: *Gr*[1] = *Gr*[3] = *Gr*[5] = 1; *Gr*[2] = *Gr*[4] = *Gr*[7] = 2; *Gr*[6] = 3.

Фрагмент основной логики.

....

*<**формирование описания графа**>;*

For i:=1 To n Do Gr[i]:=Color(i,0);

*<**вывод решения**>;*

Поиск цвета раскраски для одной вершины можно реализовать с помощью следующей функции:

Function Color(i,t:V):Integer; {*i* – номер окрашиваемой вершины, *t* – номер цвета, с которого следует искать раскраску данной вершины, *A* – матрица смежности, *Gr* – результирующий массив.}

Var Ws:Ss;

j:Byte;

Begin

Ws:=[ ];

For j:=1 To i-1 Do If A[j,i]=1 Then Ws:=Ws+[Gr[j]];{Формируем множество цветов, в которые окрашены смежные вершины с меньшими номерами.}

j:=t;

Repeat {Поиск минимального номера цвета, в который можно окрасить данную вершину.}

Inc(j);

Until Not(j In Ws);

Color:=j;

End;

Пример. На рис. 11.2 приведен пример графа. В соответствии с методом его правильная раскраска имеет вид: $Gr[1]=1$, $Gr[2]=Gr[4]=2$, $Gr[3]=3$, $Gr[5]=4$. Однако минимальной раскраской она не является: $Gr[1]=1$, $Gr[2]=Gr[5]=2$, и $Gr[3]=Gr[4]=3$, и $\chi(G) = 3$.

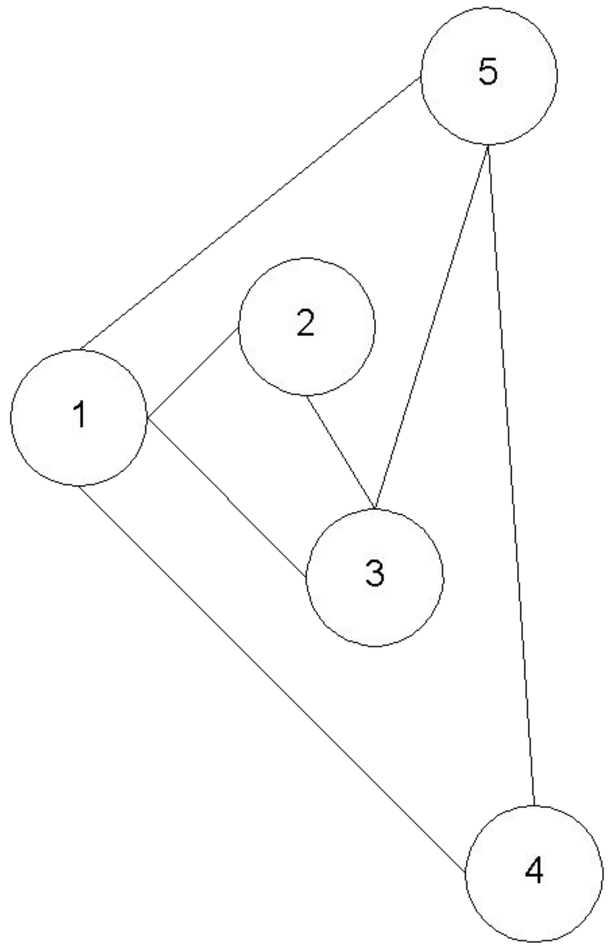

Рис. 11.2 Пример графа для иллюстрации логики нахождения правильной раскраски

Методы поиска минимальной раскраски

Метод Кристофидеса. Метод основан на простой идее, и в некоторых случаях он дает точный результат. Пусть получена правильная раскраска графа, q – количество цветов в этой раскраске. Если существует раскраска, использующая только q-1 цветов, то все вершины, окрашенные в цвет q, должны быть окрашены в цвет g, меньший q. Согласно логике формирования правильной раскраски вершина была окрашена в цвет q, потому что не могла быть окрашена в цвет с меньшим номером. Следовательно, необходимо попробовать изменить цвет у вершин, смежных с рассматриваемой. Но как это сделать? Найдем вершину с минимальным номером, окрашенную в цвет q, и просмотрим вершины, смежные с найденной. Попытаемся окрашивать смежные вершины не в минимально возможный цвет. Для этого находим очередную смежную вершину и стараемся окрасить ее в другой цвет. Если это получается, то перекрашиваем вершины с большими номерами по методу правильной раскраски. При неудаче, когда изменить цвет не удалось или правильная раскраска не уменьшила количества цветов, переходим к следующей вершине. И так, пока не будут просмотрены все смежные вершины.

Опишем логику, используя функцию *Color* предыдущего параграфа.

Procedure Solve;

Var MaxC,Num,i:V;

Begin

<ввод данных, инициализация переменных>;

For i:=1 To n Do Gr[i]:=Color(i,0); {Первая правильная раскраска.}

<найти максимальный цвет раскраски, значение переменной MaxC >;

Repeat

<найти очередную вершину, имеющую максимальный цвет раскраски(MaxC), её номер (Num)>;

<последовательно изменять цвет раскраски у вершин смежных с Num на большее значение и раскрашивать вершины с большими номерами по методу правильной раскраски >;

Until MaxC=Gr[Num] Or <все вершины с цветом MaxC просмотрены>; {До тех пор, пока не улучшим раскраску или не просмотрим все вершины с максимальным значением цвета.}

<вывод минимальной (или почти минимальной!) раскраски>;

End.

При дальнейшем уточнении логики вершину придется перекрашивать в цвет с большим значением, чем получен на предыдущих этапах. В этом случае «заработает» второй параметр функции *Color*, новое значение получается путем применения оператора *r:=Color(q,Gr[q])*, где значение *r* – новый цвет вершины с номером *q*.

Примечания. 1. При любом упорядочении вершин допустимые цвета *j* для вершины с номером *i* удовлетворяют условию $j \le i$. Это очевидно, так как вершине *i* предшествует *i*-1 вершина, и, следовательно, никакой цвет $j > i$ не использовался. Итак, для вершины 1 допустимый цвет 1, для 2 – цвет 1 и 2 и так далее.

2. С точки зрения скорости вычислений вершины следует помечать (присваивать номера) так, чтобы первые *q* вершин образовывали наибольшую клику графа *G*. Это приведет к тому, что каждая из этих вершин имеет один допустимый цвет и процесс возврата в алгоритме можно будет заканчивать при достижении вершины из этого множества.

3. Для получения первого приближения в поиске минимальной раскраски целесообразно перенумеровать вершины графа в соответствии с не возрастанием степеней вершин.

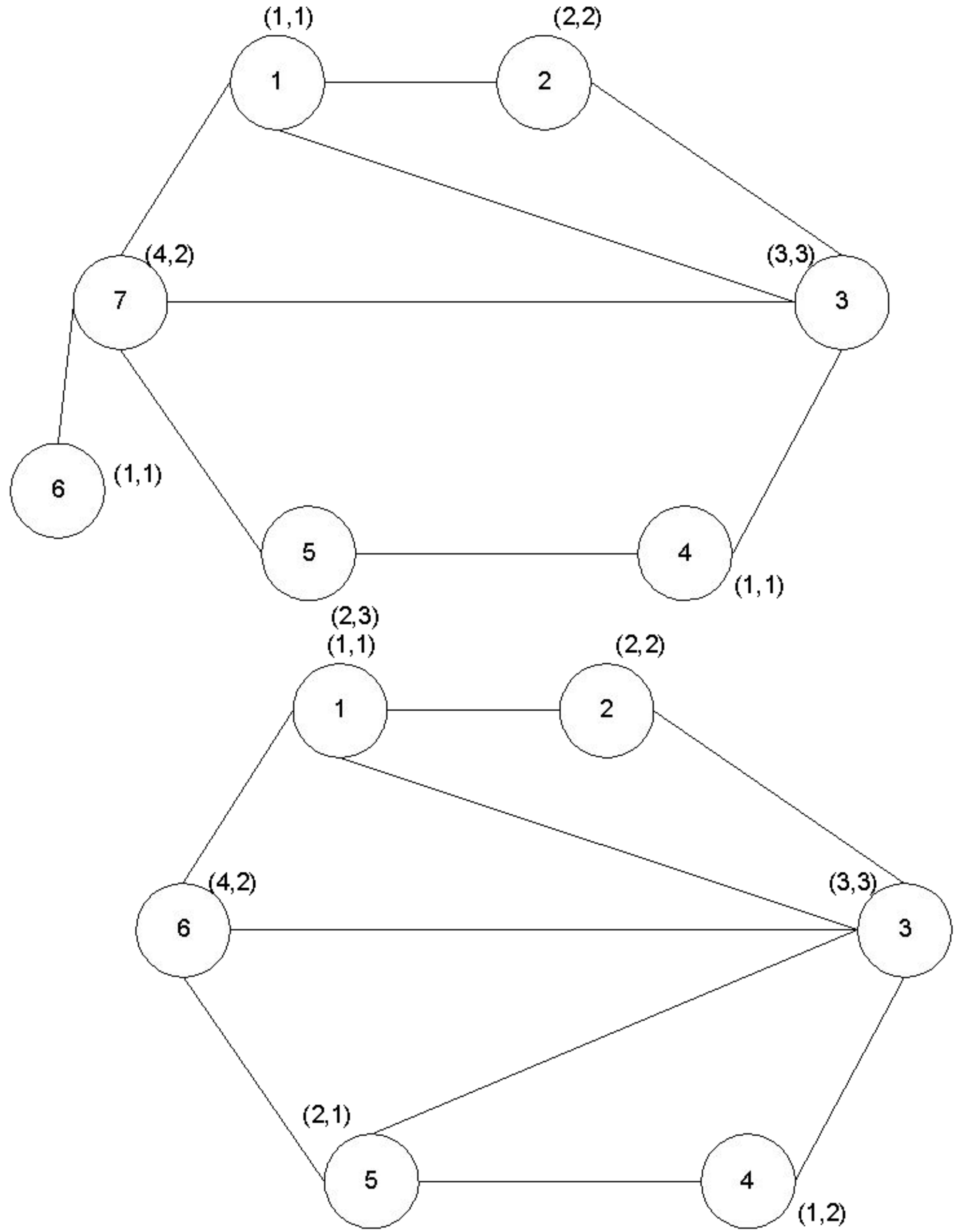

Рис. 11.3 Примеры графов. Первая цифра в круглых скобках обозначает значение цвета при правильной раскраске, вторая – при минимальной

Если для первого графа на рис. 11.3 минимальная раскраска может быть получена изменением цвета у смежной вершины с вершиной, имеющей максимальный цвет, то у второго требуется изменить цвет у смежной к смежным. Цвет изменяется у вершины с номером 4, она смежная к пятой, а не шестой, имеющей максимальный цвет.

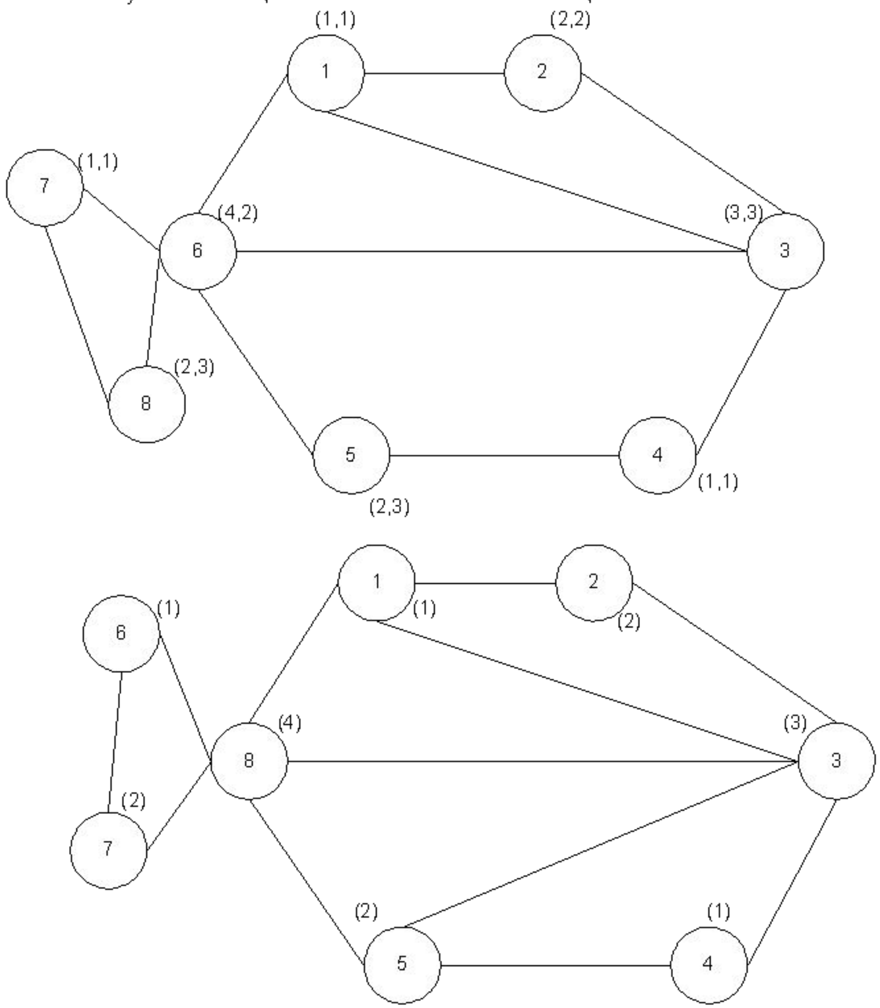

Рис. 11.4. Примеры графов. Для первого графа минимальную раскраску можно получить с помощью изменения цветов у смежных вершин, для второго – нет

На рис. 11.4 приведены два графа, они идентичны с точностью до нумерации вершин. Для первого с помощью описанного метода минимальная раскраска получается, для второго – нет, даже в том случае, когда изменяются цвета вершин у смежных к смежным. Таким образом, результат работы зависит от способа нумерации вершин графа, а таких способов $n!$.

Метод, основанный на решении задачи о покрытии. При любой правильной раскраске графа G множество вершин, окрашиваемых в один и тот же цвет, является независимым множеством. Поэтому раскраску можно понимать как разбиение вершин графа на независимые множества. Если рассматривать только максимальные независимые множества, то раскраска – это не что иное, как покрытие вершин графа G множествами этого типа. В том случае, когда вершина принадлежит не одному максимальному независимому множеству, допустимы различные раскраски с одним и тем же количеством цветов. Эту вершину можно окрашивать в цвета тех множеств, которым она принадлежит. Исходным положением метода является получение всех максимальных независимых множеств и хранение их в некоторой матрице $M(n*w)$, где w - количество максимальных независимых множеств.

$$M[i,j] = \begin{cases} 1, \text{ если вершина с номером } i \text{ принадлежит множеству с номером } j \\ 0, \text{ в противном случае} \end{cases}$$

Пример. На рис. 11.5 приведен пример графа, у которого пять максимальных независимых множеств. Матрица W имеет вид:

$$W = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Множества столбцов W [1, 3, 5] и [2, 3, 4] являются решением задачи о покрытии. Они дают два варианта минимальной раскраски вершин графа. В обоих случаях вершина 4 может быть окрашена как во второй, так и в третий цвета.

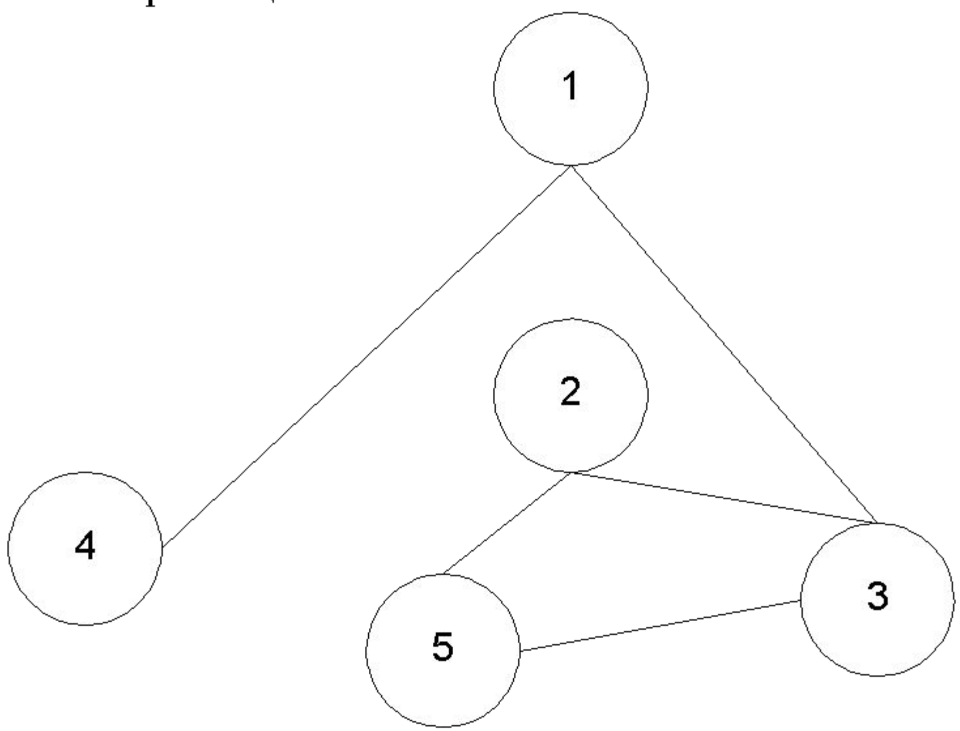

Рис. 11.5. Пример графа для иллюстрации логики сведения задачи о раскраске вершин графа