

Дискретный анализ

Соди́ков Махмуд

Разбор всех билетов к экзамену

18 сентября 2019 г.

Глава 2. Графы

1. (3) Определение графа, графов с петлями и кратными ребрами. Ориентированные графы. Соотношение между числом вершин и ребер дерева.

Определение. Графом называется пара множеств $(V, E) = G$, где V – это множество каких-то объектов, называемых вершинами графа (англ. vertices), а E – некое отношение, заданное над множеством V , то есть множество пар вершин (v_1, v_2) .

Графом с петлями назовем такую же пару множеств $(V, E) = G$, но при этом примем, что отношение E рефлексивно (то есть существует ребро из вершины в нее саму же).

Графом же с кратными ребрами (или, в последствии, мультиграфом) назовем граф, в котором две вершины могут быть соединены двумя и более ребрами.

Граф назовем *ориентированным*, если отношение E не является симметричным, то есть, говоря иначе, зададим направление (ориентацию) ребрам графа.

Деревом называется связный и ациклический граф. Перечислим основные, равносильные между собой, свойства деревьев. Положим, для начала, что G – некий граф.

0) G – дерево;

1) Между любыми 2-мя вершинами в G есть ровно один путь (простой маршрут, проходящий через различные ребра);

Док-во: (из 0-го выведем 1-ое) Если G – дерево, то граф связан. Значит, между любыми двумя его вершинами существует путь. Если найдется 2 различных пути, то найдется цикл (то есть сможем пройти от какой-то из вершин, лежащих в обоих путях, до вершины 2 и обратно до той же вершины, но через другой путь). Но по определению, дерево – граф без циклов. То есть приходим к противоречию. Значит, в дереве все пути, соединяющие вершины различны.

2) Граф G связный и количество ребер в нем на 1 меньше, чем количество вершин ($|E| = |V| - 1$);

Док-во: (из 1-го выведем 2-ое) Пусть в G между любыми 2-мя вершинами существует простой путь. Пройдем индукцией по числу вершин. Для 1-й вершины все очевидно (граф из одной вершины формально связан и количество ребер на 1 меньше, чем вершин, то есть 0). Пусть есть уже n вершин и для них верно условие. Добавим еще одну. Между любыми 2-мя вершинами существует простой путь (по условию), причем единственный. Значит, добавленная вершина соединена ровно с одной какой-то другой вершиной из прежнего графа (если с двумя и больше, то существует уже два и боль-

ше путей между новой вершиной и другими вершинами). Новый граф связан, так как прежний был связан, а значит от вершины, с которой мы соединили новую, можно было пройти до любой другой, а значит и от новой вершины можно пройти к любой другой. Количество ребер в новом графе: $|E'| = |E| + 1 = |V| - 1 + 1 = |V|$. А так как мы добавили ровно одну новую вершину, то $|V| = |V'| - 1$, а значит $|E'| = |V'| - 1$, ч.т.д.

3) *Граф G ацикличесок и количество ребер в нем на 1 меньше, чем количество вершин ($|E| = |V| - 1$);*

Док-во: (из 2-го выведем 3-е) Имеем, что граф G связан и количество ребер на 1 меньше, чем вершин. По сути, нам надо доказать, что условия ациклическости и связности, в каком-то смысле, равносильные признаки деревьев, при условии соблюдения уравнения связи между ребрами и вершинами ($|E| = |V| - 1$). Аналогично предыдущему пункту, пройдемся индукцией по количеству вершин, для одной вершины утверждение очевидно. Пусть верно для $|V|$ вершин, то есть граф на стольких вершинах ацикличесок (предположение индукции) и связан (условие), при этом выполняется соотношение между ребрами и вершинами ($|E| = |V| - 1$). Добавим новую вершину. Так как сохранилось соотношение, то мы не можем провести больше одного ребра, но обязаны провести хотя бы одно от новой вершины, чтобы сохранить связность. К какой бы вершине мы ни провели новое ребро, прежний граф был ацикличесок, а значит и новый будет ацикличесок (так как очевидным образом новое ребро не образует цикла).

0) *Граф G – дерево.*

Доказательство: (из 3-го выведем 0-е) Тоже проведем индукцию. Имеем, что для любого количества вершин граф ацикличесок и верно $|E| = |V| - 1$. Рассмотрим базу: очевидно, что одна вершина – дерево (связный ациклический граф). Пусть для некоторого количества вершин $|V|$ верно, что это к тому же и дерево (то есть связано + наше условие). Выведем, что он ацикличесок. при добавлении одного ребра. Опять-таки, мы обязаны добавить ровно одно ребро, чтобы выполнить соотношение. Но мы можем добавлять ребро и между вершинами старого графа. Но тогда, заметим, что любое новое ребро в старом графе нарушит ациклическость, так как старый граф был деревом, значит выполнялся пункт 2), а значит, между любыми 2-мя вершинами был ровно один (!) путь. Добавление в такую конструкцию нового ребра где бы то ни было, приведет к появлению цикла, так как теперь мы от вершины v_1 к вершине v_3 можем добраться через промежуточную вершину v_2 либо через вершину, которая через новое ребро соединяет некую вершину v_2' и от которой можно по другому пути добраться до вершины v_3 . А значит, новое ребро соединяет новую вершину и какую-то из вершин старого графа. А значит, наш граф связан, значит имеем дерево.

2. (5) Код Прюфера. Формула Кэли.

Начнем с того, что определим метод кодирования Прюфера, как некий алгоритм. Пусть у нас есть дерево, с некоторой нумерацией вершин.

Утверждение без доказательства состоит в том, что в этом дереве есть хотя бы одна висячая вершина, то есть вершина со степенью 1, или т.н. лист (если угодно, можно сказать, что это очевидно в силу банальной индукции по количеству вершин: (база очевидна – дерево с одной вершиной имеет висячую вершину) пусть есть дерево, в котором есть висячая вершина и мы добавляем новую вершину. Тогда мы либо соединим ее с невисячей вершиной, то есть прежняя висячая вершина так и останется висячей в новом дереве, либо мы соединим новую вершину с листом, и сделаем новую вершину саму висячей, так как ее степень очевидно равна 1).

Теперь покажем алгоритм кодирования Прюфера: пусть у нас дерево на n вершинах. Пусть имеется также строка S – наш будущий код.

- 1) *Найдем висячую вершину с наименьшим номером;*
- 2) *Удалим ее, заранее записав номер вершины, в которую ведет этот лист, в конец нашего кода S ;*
- 3) *Повторим пункты 1) и 2) с нашим новым деревом, пока не получим дерево с одним ребром (на двух вершинах);*
- 4) *Получившуюся строку S и назовем кодом Прюфера.*

Сразу покажем алгоритм декодировки. Пусть на плоскости есть только вершины от 0 до $|S| + 2$

- 1) *Напишем под кодом S числа от 0 до $|S| + 2$;*
- 2) *Найдем в нижней последовательности минимальное число k , не входящее в S ;*
- 3) *Соединим вершину с номером $S[0]$ и k ребром в графе;*
- 4) *Повторим пункты 1) - 3), пока не получим в нижней последовательности два числа, вершины с номерами которых также соединим. Получим исходный граф.*

На базе этого нехитрого метода кодирования графов можно доказать, что количество деревьев на n вершинах равно $t_n = n^{n-2}$. Но прежде необходимо обосновать корректность кодирования Прюфера, то есть показать, что в принципе этот алгоритм можно назвать кодированием, в классическом понимании этого термина, иначе говоря, показать биекцию между множеством кодов Прюфера и множеством всех деревьев.

Этот факт даст нам однозначность кодирования-декодирования, а значит, посчитав количество слов (кодов) Прюфера длины n мы сможем посчитать и количество деревьев на n вершинах (в силу биективности эти две величины будут равны).

Начнем с доказательства того, что алгоритм Прюфера является инъекцией.

Сначала уясним факт, что вершина с номером i встречается в коде Прюфера ровно на 1 меньше, чем ее степень. Это очевидно, так как по построению кода Прюфера, мы записываем i -ую вершину в код каждый раз, когда удаляем ребро, ведущее из нее в вершину, которая на данной итерации оказалась листом, то есть каждый раз, когда удаляем все ребра, которые соединяют i -ую вершину с наименьшими вершинами. До того, как мы удалим вершину с номером i , она не будет являться листом. Как только она станет листом, есть шанс, что она может быть удалена на i -ой итерации. Если она удаляется, то мы ее номер встретим ровно $\deg(v_i) - 1$ раз. Если мы эту вершину так и не удалили, значит она оказалась среди одной из последних двух оставшихся, а значит, мы все равно встретим ее номер ровно $\deg(v_i) - 1$ раз. Логика и идея доказательства в том, что мы не удалим вершину с номером i , пока не удалим ровно $\deg(v_i) - 1$ инцидентных ей, а равно, исходя из имплементации алгоритма, пока не запишем номер вершины в наш код ровно столько же раз.

Теперь, пройдемся индукцией по количеству вершин графа. При $|V| = 2$ примем очевидным тот факт, что код Прюфера задается однозначно для любой разметки вершин дерева с одним ребром. Теперь пусть имеется дерево на n вершинах. Пусть оно удовлетворяет однозначности кодирования методом Прюфера (для любой разметки). Тогда рассмотрим два дерева, у которых $n+1$ вершин и которые отвечают одному и тому же коду Прюфера (вдруг так получилось). Тогда посмотрим на листы с минимальными номерами в обоих деревьях. Если они не совпадают, то из предыдущего утверждения, не совпадают коды, так как каждая вершина встречается $\deg(v_i) - 1$ раз (то есть если вершина под номером i является листом в первом дереве, а во втором не является, значит степень этой вершины во втором дереве не равна нулю, в отличии от первого, то есть номер i входит в код Прюфера второго дерева, но не входит в код Прюфера первого). Допустим теперь, что номера листов совпали. Тогда удаляем листы в обоих графах и получаем дерево на n вершинах, которое по предположению индукции однозначно кодируется методом Прюфера. Получили, таким образом, что любое дерево можно закодировать методом Прюфера, причем однозначно (инъективность).

Теперь покажем, что любой код Прюфера можно корректно преобразовать в дерево (сюръекция). Оттуда и получим биекцию.

Опять применим индукцию. Для $n = 1$ все очевидно (код Прюфера будет задавать дерево длины 3, причем сам код будет в точности обозначать ту вершину, которая будет "посередине" в этом дереве (то есть иметь степень 2)). Далее, пусть имеем код S Прюфера длины n и для него умеем определять корректно соответствующее дерево (которое, кстати, при обратной кодировке будет давать нам в точности исходный код Прюфера). Теперь рассмотрим код Прюфера длиной $n + 1$. Тогда найдем из $n + 3$ чисел число v ,

которое минимально и не входит в S . По нашей лемме о степенях, это будет как раз тот лист, который при кодировке мы удаляли первым. Соединим его с $S[0]$. Удалим нулевой элемент строки S . В строке S декрементируем все элементы, большие чем v и обозначим их штрихом. Получим задачу декодирования кода Прюфера длиной n . По предположению индукции, это возможно, причем корректно. Следовательно, мы доказали теорему об однозначном кодировании/декодировании кодов Прюфера.

Теперь, наконец-то, можем посчитать количество деревьев на n вершинах. Так как мы установили биекцию между словами длины $n-2$, в каждой позиции которых может стоять любая цифра от 1 до n , то нетрудно догадаться, что количество деревьев на n вершинах абсолютно точно равно количеству таких слов. Совершенно очевидно, что таких слов как раз $t_n = n^{n-2}$.

3. (6) Точная формула для числа унициклических графов.

Граф G назовем унициклическим, если в нем имеется ровно 1 цикл и он связан, или, что равносильно, количество ребер равно количеству вершин и он связан. То есть отличие дерева от унициклического графа в том, что унициклический граф имеет на 1 ребро больше (которое как раз и образует тот самый единственный цикл, ввиду того, что дерево является минимальным по числу ребер связным графом среди всех связных графов на данных n вершинах).

Обозначим за U_n количество унициклических графов на n вершинах. Попытаемся найти их точное количество.

Будем строить нашу конструкцию вокруг того самого единственного цикла графа. Обозначим за r количество вершин, входящих в этот цикл. Вполне очевидно, что $r \geq 3$.

Унициклический граф устроен так, что вокруг этого цикла развито несколько (что важно, не имеющих общих вершин) деревьев. Иначе говоря, лес на r вершинах. Причем, вполне возможно, что деревья в этом лесу вырожденные (то есть состоят из 1 вершины).

Вариантов выбрать r вершин равно $C_n^r = \frac{n!}{(n-r)!r!}$. Но нас интересует еще и нумерация вершин в этом цикле. А значит, нам нужно для каждого варианта выбора вершин цикла выбрать еще и $\frac{(r-1)!}{2}$ способов расстановки номеров. Кроме того, нам нужно еще умножить все это на количество лесов на n вершинах из r деревьев, то есть на $F_n = rn^{n-1-r}$. Таким образом, получаем, что точное количество унициклических графов на n вершинах

$$U_n = \sum_{r=3}^n C_n^r \frac{(r-1)!}{2} rn^{n-1-r}$$

4. (5) Определение плоских и планарных графов. Формула Эйлера. Примеры непланарных графов. Критерий Портягина-Куратовского (б/д)

Граф называется планарным, если существует его укладка в плоскости \mathbb{R}^2 . Соответственно, укладкой графа мы назовем такие непрерывные преобразования, которые бы переводили граф в плоскость \mathbb{R}^2 , причем желательно сохраняя свойство изоморфизма двух графов. В зависимости от той точности, которая стоит за словом изоморфизм, можно вводить различные методы укладки графов. Мы же возьмем следующий:

Назовем укладкой графа $G = (V, E)$ пару преобразований (F, H) , такую, что:

$$F : V \rightarrow S, \quad S \subset \mathbb{R}^2, \quad |S| < \infty$$
$$H : E \rightarrow W,$$

где W – это множество гладких кривых, в которые преобразуются ранее введенные "абстрактные" ребра графов. На преобразование H накладывается дополнительно еще то условие, что множество кривых такое, что они соединяют два образа вершин, которые задает F только тогда, когда имеется ребро между этими вершинами. Проще говоря, модель (F, H) является формализацией стандартной для нас операции изображения графа на листе бумаги.

Будем называть класс графов, которые можно нарисовать без самопересечений гладких кривых из W , которые соединяют образы вершин на плоскости, *плоскими графами*. Тогда граф будем называть *планарным*, если он изоморфен некоторому плоскому графу.

Для **связного** плоского графа справедлива формула Эйлера:

$$n - e + f = 2,$$

где $|E| = e$, $|V| = n$, а f – количество *граней* графа. Гранью укладки назовем область плоскости, которая отделена ребрами графа (и не содержит внутри себя других ребер). И обратно, связный граф является планарным, если удовлетворяет формуле Эйлера.

Доказательство: Классическая индукция. На этот раз по разности $e - n$.

Рассмотрим базу: $e - n = -1 \Rightarrow$ наш граф является деревом, так как он связан из условия выполнимости критерия Эйлера и у него ребер на 1 меньше, чем вершин. Очевидно, что любое дерево планарно, так как в дереве нет даже циклов, то есть мы точно сможем нарисовать на плоскости любое дерево без самопересечений образов-кривых ребер. И для любого дерева существует ровно одна грань (внешняя) других нет за отсутствием циклов в дереве. Значит, выполняется формула Эйлера.

Сделаем предположение и шаг: Пусть для любого плоского графа, у которого $e - n = k$ выполняется формула Эйлера. Попробуем увеличить разность и рассмотреть графы у

которых $e - n = k + 1$. Имеем некоторый связный граф, который точно не является деревом, а значит в нем есть циклы. Любой цикл отделяет две различные грани (в крайнем случае грань, образованную циклом от внешней грани). Удалим какое-нибудь ребро в этом цикле. Тогда количество граней тоже уменьшится на 1. Пусть у нового графа теперь e' ребер и f' граней. Количество вершин не изменилось, а значит наш граф из класса графов предположения индукции. Тогда верно, что $n - e' + f' = 2 = n - (e - 1) + (f - 1) = n - e + 2$. ч.т.д.

Теперь рассмотрим примеры непланарных графов. Прежде, чем это сделать, введем соответствие между графами, называемое гомоморфизмом графов.

Два графа гомеоморфны, если один может быть получен из другого любым набором из следующих преобразований:

- 1) Изоморфизмом двух графов (то есть любым преобразованием, сохраняющим изоморфность);
- 2) Разбиение ребра (добавлением средней вершины);
- 3) Стягивание вершины.

Графы K_5 и $K_{3,3}$ являются непланарными.

Критерий планарности графов (Портнягина-Куратовского): Граф планарен, тогда и только тогда, когда он не содержит подграфа, гомеоморфного графам K_5 или $K_{3,3}$.

6. (5) Последовательности и графы де Брёйна

Последовательность де Брёйна – такая последовательность из алфавита $\{0, 1, \dots, n\}$, которая содержит в качестве своих подбуферов (отрезков) длины k все последовательности длины k из $n+1$ элементов. Например, одна из последовательностей при параметрах $k = 2$ и $n = 1$ имеет вид 0011 и содержит подпоследовательности 00, 01, 11, 10 длины 2 из алфавита $\{0, 1\}$.

Используется классическое правило "0 лучше 1" для построения последовательностей де Брёйна: изначально ставим k единиц, далее, пока можем, ставим нули. Иначе 1, пока не получим из подбуферов все подмножества. Так как всего 2^k таких слов, то длина последовательности де Брёйна равна $2^k + k - 1$. Это ясно, если представить последовательность как таблицу.

В общем же случае, длина последовательности де Брёйна с длиной буфера k и размером алфавита n будет равно $n^k + k - 1$.

Граф де Брёйна – орграф, который в качестве вершин имеет все подмножества длины k , а ребром две вершины соединяются, если конец строки на одной вершине равен началу на другой (ориентация от первой вершины ко второй).

7. (5) Гамильтоновы пути и циклы. Достаточное условие Дирака гамильтоновости графа.

Граф называется гамильтоновым, если он является ребернопростым циклом.

Гамильтонов путь – (реберно)простой путь, проходящий через все вершины графа.

Гамильтонов цикл – замкнутый гамильтонов путь (или ребернопростой цикл, проходящий через все вершины графа).

Достаточное условие Дирака: Если в связном графе степень каждой вершины $\geq \frac{n}{2}$, то граф гамильтонов.

Доказательство: Пусть $P = v_1 v_2 \dots v_k$ – самый длинный путь в графе G . Если v_1 смежна с некоторой вершиной $x \notin P$, то существует путь длиннее P – противоречие. Аналогично рассуждение с $v_k \Rightarrow v_1$ и v_k смежны только с вершинами из P . Поскольку $\deg(v_1) \geq \frac{n}{2}$ и в графе нет петель, то $k \geq \frac{n}{2} + 1$.

Теперь докажем, что в графе есть такая вершина v_j , которая смежна с v_k , а вершина, инцидентная ей и лежащая в нашем пути $P - v_{j+1}$ – смежна с v_1 . Действительно, это так, потому что иначе в нашем графе есть ровно столько вершин, несмежных v_k , сколько вершин инцидентны v_1 , так как все их предыдущие в нашем пути соседи должны быть не смежны с v_k . Значит, в P хотя бы $\deg(v_1) + \deg(v_k) + 1 = n + 1$ вершин. Противоречие. Значит, в G возникает простой цикл, проходящий по нашим найденным ребрам из конечных точек пути P и через $P/(v_j, v_{j+1})$

И теперь, так как путь P самый длинный в нашем графе, то мы не можем его увеличить. В то же время, наш граф по условию связан, а значит, любая вершина обязана иметь ребро с каким-то из вершин в нашем цикле. Очевидно, что из этого следует, что всякая вершина графа лежит в нашем цикле и в нашем пути P . Этот цикл и будет гамильтоновым.

8. (7) Вершинная связность и число независимости. Достаточное условие гамильтоновости в их терминах. Гамильтоновость графа $G(n, 3, 1)$

Определение: Вершинная связность $k(G)$ графа – минимально количество вершин, в результате удаления которых граф перестает быть связным.

Определение: Число независимости графа $G = (V, E)$

$$\alpha(G) = \max\{|W| : W \subset V, \forall v_i, v_j \in W \rightarrow (v_i, v_j) \notin E\}$$

Достаточное условие гамильтоновости (Эрдеша-Хватала): Если в графе $G = (V, E)$ $|V| \geq 3$ $\alpha(G) \geq k(G)$, то граф гамильтонов.

Доказательство: Пусть $|V| = n \geq 3$. Пусть в G нет циклов для начала. Тогда число независимости больше 1, значит $k(G) \geq 1$. Значит, граф связан, так как нужно удалить более одной вершины, чтобы граф перестал быть связным. Значит, это дерево, так как нет циклов и граф связный. Тогда, в дереве есть хотя бы 2 висячие вершины, то есть $\alpha(G) \geq 2$, но это противоречит условию.

Далее, рассмотрим случай, когда цикл есть. Найдем самый длинный простой цикл. Пусть он не исчерпывает все вершины (то есть он не гамильтонов). Придем к противоречию.

Удалим этот цикл со всеми инцидентными ребрами. Останется некий граф. Пусть W – любая связная компонента оставшегося графа. Рассмотрим множество соседей всех вершин из этой компоненты. Обозначим его за $N(W)$.

(Далее картинки в тетради, включу сюда позже). Общий план: рассматриваем два случая: когда из цикла есть две вершины подряд из которых есть ребро в W , доказываем, что этих ребер не может быть, далее рассматриваем тот же случай, но когда есть ребро между двумя вершинами внутри цикла. В обоих случаях получаем, что наше множество является независимым, что невозможно. Рассматриваем мощности и приходим к противоречию.

Для графа $G(n, 3, 1)$: признак Дирака не применим, так как степень каждой вершины примерно квадратична, так как равна C_{n-3}^2 (соединена с двумя вершинами из множества без 3-х вершин)

13. (4) Соотношения между хроматическим числом, числом независимости и кликовым числом графа.

Хроматическое число равно отношению количества ребер на число независимости. Действительно, по факту – хроматическое число разбивает множество вершин графа на χ независимых подмножества.

$$|V| = \sum_{i=1}^{\chi(G)} |V_i| \leq \chi |V_{max}| = \chi \alpha(G)$$

Также, хроматическое число не меньше кликового числа, так как надо хотя бы $\omega(G)$ цветов, для того, чтобы раскрасить максимальную клику в графе G .

14. (3) Числа Рамсея. Рекуррентная верхняя оценка Эрдеша-Секереша. Точные значения для $s+t \leq 7$.

Числом Рамсея назовем следующую величину:

$R(s, t) := \min\{\text{число } n, \text{ такое, что любая раскраска полного графа на } n \text{ вершинах имеет подклику первого (синего) цвета размера } s, \text{ либо имеют подклику второго (красного) цвета размера } t\}$

Верна следующая рекуррентная оценка Эрдеша-Секереша:

$$R(s, t) \leq R(s-1, t) + R(s, t-1)$$

Для доказательства, зафиксируем какую-то раскраску ребер и какую-то вершину v , при этом изначально положим, что граф состоит из $n = R(s-1, t) + R(s, t-1)$ вершин (граф полный). Теперь рассмотрим две альтернативы: из вершины v выходит не более $R(s-1, t)$ ребер, соединенные красным ребром, либо выходит не более $R(s, t-1)$ синих ребер. Это исчерпывающие друг друга альтернативы, так как если бы не выполнялись они оба, то есть из нашей вершины выходило бы не более $R(s-1, t) - 1$ красных и сразу не более $R(s, t-1) - 1$ синих, то всего выходило бы не более $R(s-1, t) - 1 + R(s, t-1) - 1 = n - 2$ ребер, что, конечно не так.

Рассмотрим первый вариант, второй аналогичен. Если выходит столько вершин, то значит есть среди этих вершин, так как граф полный, синяя клика размера t , либо красная клика размера $s-1$. Если нашлась красная клика, то мы соединяем все ребра, идущие от нашей вершины v (они красного цвета) и они образуют ровно s -клику, которая нам и была нужна.

Значит, мы во всем графе нашли $R(s, t)$, а значит, нам было бы достаточно количества вершин, меньших $R(s-1, t)$ вершин.

Разбирая вторую альтернативу, приходим к тому, что $R(s, t)$ не больше заявленной суммы.

15. (4) Следствия из верхней оценки чисел Рамсея. Уточнение Конлона(б/д). Нижняя оценка с помощью вероятностного метода.

$$R(s, t) \leq C_{s+t-2}^{s-1}$$

Доказательство: Вспомним из курса ОКТЧ равенство $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$.

И проведем аналогию, между доказанной ранее рекурсивной оценкой и этим равенством. В качестве n возьмем $s+t-2$.

Интереснее рассмотреть нижнюю оценку диагональных чисел Рамсея. Для его доказательства пользуемся вероятностным методом и доказываем, что для графа с $n = (1 + o(1)) \frac{1}{e\sqrt{2}} s 2^{\frac{s}{2}}$ возможно подобрать такую раскраску, что не найдется клик одного цвета размера s . Для этого рассматриваем вероятность того, что объединение клик по s вершин раскрашено в один цвет. Получаем, что вероятность этого события не больше 1, а значит, вероятность отрицания события больше 0, то есть вероятность пересечений отрицания каждого события больше 0, то есть можно так раскрасить граф, чтобы в нем не было таких клик. (Это число в оценке следует из сокращений и упрощений в конечной формуле вероятности, о малое нужно для того, чтобы не нарушить "асимптотичность" при малых s , иначе говоря, можем сказать, что неравенство верно без скобки с o малым начиная с какого-то s).

16. (5) Многоцветные числа Рамсея и их рекуррентная верхняя оценка. Следствие для $R_3(s, t)$. Нижняя оценка для $R_3(s, s)$.

Многоцветное число Рамсея $R(m_1, m_2, \dots, m_k)$ – минимальное из таких чисел x , что для любой раскраски ребер полного графа на x вершинах, для некоторого i найдется клика на m_i вершинах. (Будет дополняться)

22. (5) Верхняя оценка размера минимальной с.о.п. с помощью жадного алгоритма.

Пусть дано множество $\mathcal{R}_n = \{1, 2, \dots, n\}$.

Рассмотрим множество $\mathcal{M} = \{M_1, M_2, \dots, M_s\}$, такое что $M_i \subset \mathcal{R}_n$. Система общих представителей для \mathcal{M} любое множество $S \subset \mathcal{R}_n : \forall i M_i \cap S \neq \emptyset$.

Введем также переменную $\tau(\mathcal{M}) = \min\{|S|, S - \text{с.о.п.}\}$

Жадный алгоритм заключается в следующем: мы будем на каждой итерации брать максимальный по вхождению элемент и добавлять его в с.о.п., параллельно "вычеркивая" множества, в которые этот элемент входил (иначе говоря, с.о.п. будет точно их покрывать). Как только покрывли все множества в \mathcal{M} , останавливаем алгоритм. Полученное множество S и есть с.о.п.

Покажем для начала тривиальные оценки числа τ . Вполне понятно, что минимальное количество элементов в с.о.п. не превышает $\min s, n-k+1$. Действительно, оно не может быть больше, чем все количество множеств (то есть если мы возьмем даже по 1 элементу из каждого множества, мы максимум получим с.о.п. мощности s). Также ясно, почему оно не больше чем $n-k+1$: пусть мы возьмем все первые $n-k+1$ из \mathcal{R}_n . Тогда, как бы мы ни брали наши множества из \mathcal{M} , не найдется множества мощности k в \mathcal{R}_n , которое с ними не пересекается (даже последние k элементов будут пересекаться по 1 элементу).

Тривиальной оценкой снизу является то, что $\tau \geq \min\{s, \lfloor \frac{n}{k} \rfloor\}$. Если $s \leq \lfloor \frac{n}{k} \rfloor$, то просто будем брать непересекающиеся множества для системы $\lfloor \frac{n}{k} \rfloor$ штук точно хватит (если будет меньше, то ровно s , как раз) и возьмем по 1 элементу из каждого множества такой системы. Если количество множеств системы больше, чем $\lfloor \frac{n}{k} \rfloor$ ($s \geq \lfloor \frac{n}{k} \rfloor$), то тогда еще более тривиально – просто возьмем все $\lfloor \frac{n}{k} \rfloor$ штук, а остальные насыпаем "сверху" кастрюльки.

Теперь оценим количество элементов соп, полученное жадным алгоритмом.

Теорема: $\tau(\mathcal{M}) \leq \max\{\frac{n}{k}, \frac{n}{k} \ln(\frac{sk}{n})\} + \frac{n}{k} + 1$

Доказательство: случаи, когда $s \leq \lfloor \frac{n}{k} \rfloor$ и $\frac{n}{k} \ln(\frac{sk}{n}) \geq n$ разобраны ранее.

Перейдем к содержательной ситуации, когда $s > \lfloor \frac{n}{k} \rfloor$ и $\frac{n}{k} \ln(\frac{sk}{n}) < n$.

Будем пользоваться жадным алгоритмом. Найдем элемент, максимальный по вхождению во все множества нашей системы \mathcal{M} . Количество элементов системы M_i , которые имеют своим элементом найденный алгоритмом обозначим p_1 .

Если рассматривать модель, будто наши множества из \mathcal{M} лежат в некоторой таблице (где по оси y высота этой модели как раз равна p_1 , а по оси x у нас числа от 1 до n), то так как есть "пустые" части между множествами из \mathcal{M} в нашей укладке, можно вывести неравенство $p_1 * n \geq s * k$, так как $s * k$ – общее количество задействованных

элементов в каждом из множеств.

Исходя из имплементации той же модели и полагая, что мы действуем в случае, когда $\frac{sk}{n} < n$, получаем, что мы сделаем жадным алгоритмом в итоге $N = \lfloor \frac{n}{k} \ln(\frac{sk}{n}) \rfloor + 1$ шагов.

Далее упрощениями и оценкой числа s_N показываем, что оно не превосходит $\lfloor \frac{n}{k} \rfloor$ и это означает, что для оставшегося после работы алгоритма множества можно просто взять по 1 представителю из каждого множества системы. Откуда и наша оценка, заявленная в формулировке.