Генерация графов «Враги в разных комнатах»

Хаханов Тимофей, 6372

Изначальная задача

Построить графы к этим задачам:

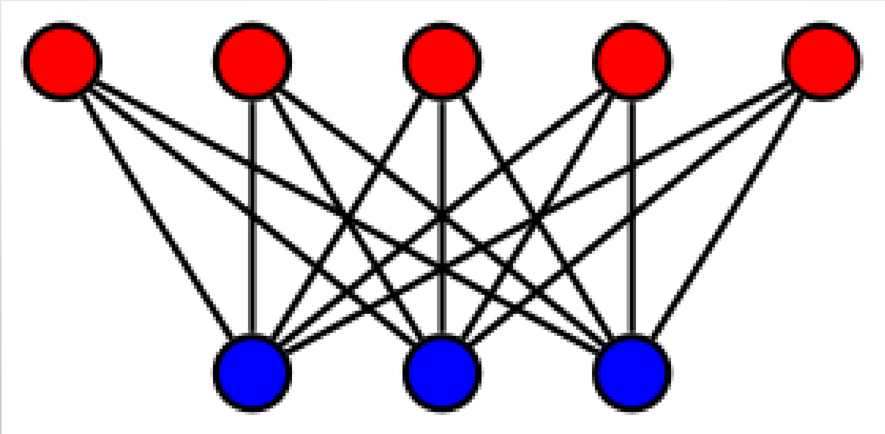
Цыбышев А.

15) Сколько максимально может быть ребер в k-дольном графе на n вершинах? (легенда: вершины — дети, соединены ребром если враги, их можно рассадить по k комнатам)
А решение такое: вычисление и варьирование (ответ - в долях должно быть как можно более поровну вершин)

16) Доказать, что граф на n вершинах с максимально возможным количеством ребер, не содержащий в качестве подграфа полного графа на $k+1$ вершине, можно правильно раскрасить в k цветов. (легенда: вершины — дети, соединены ребром если враги, "ситуация не k-тастрофическая", если среди любых $(k+1)$ детей есть невраждующая пара.)
Решение: Надо доказать индукцией по n, что любой граф с максимальным количеством ребер, не содержащий полного п.г. на $(k+1)$ вершине, изоморфен графу из пункта (1). Для перехода индукции надо удалить из графа на n вершинах вершину минимальной степени, и оценить количество ребер через количество ребер в подграфе на $(n-1)$ вершине. (база индукции — $n = (k+1)$)
В общем если посчитать, получается оценка $E(n) \le (n/(n-2)) \cdot E(n-1)$, соответственно, если $E(n-1)$ не максимально возможное, то и $E(n)$ тем более (с учётом того что для максимально возможного значения $E(n-1)$ получается $E(n) = [(n/(n-2)) \cdot E(n-1)]$ (целая часть) — это равенство проверяется подстановкой формул из пункта (1).

Что такое k-дольный граф?

- **k-дольный граф** — это граф, вершины которого можно условно поделить на доли («группы» вершин), что не существует ребра, соединяющего вершины из одной доли.
- **Полный** k-дольный граф — это такой k-дольный граф, что каждая вершина соединена ребрами со всеми вершинами графа, кроме вершин «своей» доли. Именно с этими графами мы будем работать.



Полный *двудольный* граф на *восьми* вершинах

Переход к фактической задаче

Графы второй задачи, в соответствии с доказательством, приложенным к ней, являются изоморфными первой. Поэтому для выполнения задания достаточно построить графы из первой задачи: они будут аналогичны необходимым во второй, и выбрать среди них граф с максимальным числом ребер.

Фактическая задача

- Сгенерировать все возможные полные k-дольные графы на n вершинах
- Выбрать из них граф с максимальным числом ребер

Подход к решению задачи

Задача была решена следующим образом.

Рассмотрим доли графа как «комнаты». Всего у нас k комнат, в каждой n детей. При таком подходе видно, что перед нами практически тривиальная **комбинаторная** задача: перебрать все варианты расположения «детей» по «комнатам» таких, что в каждой комнате находится хотя бы один «ребенок». При этом относительное расположение («порядок на этаже») нас не интересует, важно лишь распределение детей, т.е. необходимо **отбросить изоморфные** варианты.

Алгоритм распределения вершин

Все операции производились с массивом целых чисел длиной k, сумма элементов всегда равнялась n.

Перебор осуществлялся следующим образом:

- Создавался массив из k элементов {n-k+1, 1, …, 1}
- Рекурсивно вызывалась функция, отнимающая единицу у ведущего, и прибавляющая единицу к следующему наибольшему, но такому, что ведущий элемент не будет меньше или равен следующещему-1. Следующие запуски вызывались со смещением ведущего вправо.
- Результаты записывались в память, формируя коллекцию графов, удовлетворяющих условию.

Пример работы алгоритма:

6111->5211->4311->4221->3321->3222 – все уникальные комбинации без изоморфности

Алгоритм распределения вершин

```
static int[] Gen(int[] arr, int offset = 0)
{

    if (offset + 1 < arr.Length)
    {
        while (true)
        {
            bool wasLess = false;

            for (int i = offset + 1; i < arr.Length; ++i)
                if (arr[offset] - 1 > arr[i])
                {

                    arr[offset]--;
                    arr[i]++;
                    Graphs.Add(new GraphMap(arr));
                    arr = Gen(arr, offset + 1);
                    wasLess = true;
                    break;
                }
            if (!wasLess)
                return arr;
        }
    }
    return arr;

}
```

Максимальное число ребер

У графа тем больше ребер, чем больше сумма степеней вершин этого графа. Добиться максимальной суммы степеней в k-дольном графе можно только при максимально-равномерном распределении вершин по долям.

Т.к. алгоритм распределения вершин между долями перебирает от самого неравномерного к самому равномерному (6111->5211->4311->4221->3321->3222), максимальное число связей будет в последнем графе, сгенерированном алгоритмом (3222 в данном случае).

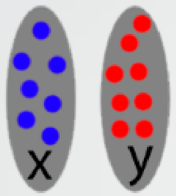
Доказательство

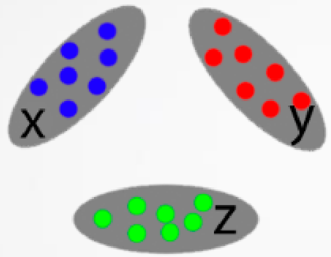
Лемма о рукопожатиях гласит, что сумма степеней вершин графа равна удвоенному числу ребер => нам необходимо добиться максимальной суммы степеней вершин.

Степень каждой вершины в k-дольном графе – число вершин, не входящих в ее долю («врагов»). Т.к. мы «распределяем» ограниченное кол-во вершин (не можем менять их число), от для того, чтобы добиться максимума на каждой доле, необходимо, чтобы каждая доля была наиболее близка к минимуму(меньше «своих» => больше «врагов»), что наблюдается при равномерном распределении (все числа уменьшены настолько, насколько возможно). Уменьшение суммы степеней при «менее равномерном» перераспределении доказывается алгебраически, рассмотрим примеры 2- и 3-дольных графов (где легче следить за вычислениями).

Расчеты изменения суммы степеней для 2- и 3-дольных графах при перераспределении

Пусть x, y, z - число вершин на доле, тогда число вершин в равномерном графе определяется так (a = x = y= z):

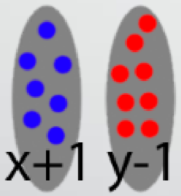


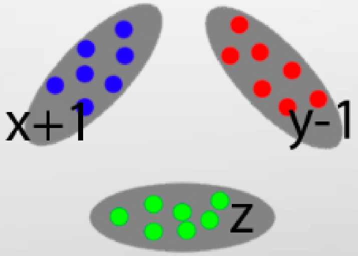


$\Sigma deg(v) = 2(x^*y) = 2a^2$

$\Sigma deg(v) = x(z+y)+y(x+z)+$
$+z(x+y) = 3a(a+a)=6a^2$

Перераспределение вызовет уменьшение степени:





$\Sigma deg(v) =$
$(x+1)(y-1)+(y-1)(x+1) =$
$= 2(xy-x+y-1) = 2(a^2-1)$

$\Sigma deg(v) = (x+1)(y+z-1)+$
$+(y-1)(x+z+1)+z(x+y+1-1)$
$= (a+1)(2a-1)+(a-1)(2a+1)+2a^2 =$
$= 2a^2-a+2a-1+2a^2+a-2a-1+2a^2 =$
$= 6a^2-2$

$2a^2 > 2(a^2-1)$

$6a^2 > 6a^2 - 2$

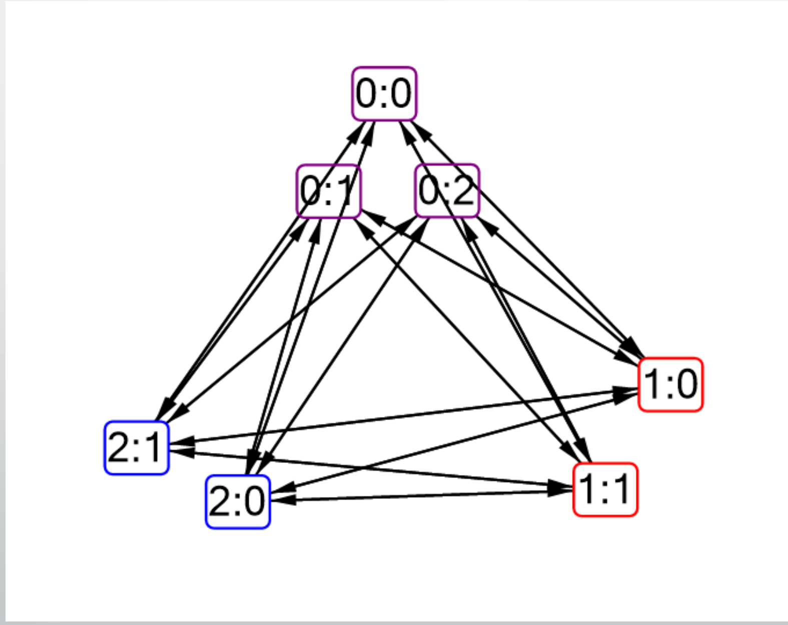
Сумма степеней при равномерном распределении определяется формулой: $\Sigma deg(v) = k*a*(k-1)$, где k – число долей, a – число вершин на доле.

Вычисления для графов на больших долях (или с большим смещением) аналогичны, общая формула следующая:
$\Sigma deg(v) = \sum_k (a_k * \sum_{j!=k}^{k} a_j)$
Где k – число долей, a_i - число вершин в i-той доле

Результат

Преобразуя полученные массивы в графы, можно получить все уникальные (не изоморфные) графы, удовлетворяющие условию, для любых n и k (n >= k). 197.703 графа (1000 долей на 100.00 вершинах) программа генерирует за 1870 **милли**секунд, при небольших входных параметрах – за доли миллисекунды (все вычисления производились на среднем, потребительском процессоре семилетней давности).

Пример сгенерированного графа:



Использованные технологии:

- Язык программирования: C# 6.0 + .NET Framework 4.5
- IDE: Microsoft Visual Studio 2015
- Интерфейс: Windows Forms
- Графическая библиотека для вывода графов: MsAlg Automatic graph layout
- Презентация: Microsoft PowerPoint