

МНСК2024

Гармонические разметки вершин графа

Шуйская Ярослава
11-1 класс СУНЦ НГУ

Научный руководитель -
к.ф.-м.н. С. В. Августинovich



Порядок выполнения исследования

Задача про
рыцарей и
лжецов

01

02

Решение задачи в
терминах
гармонических
разметок для всех
деревьев

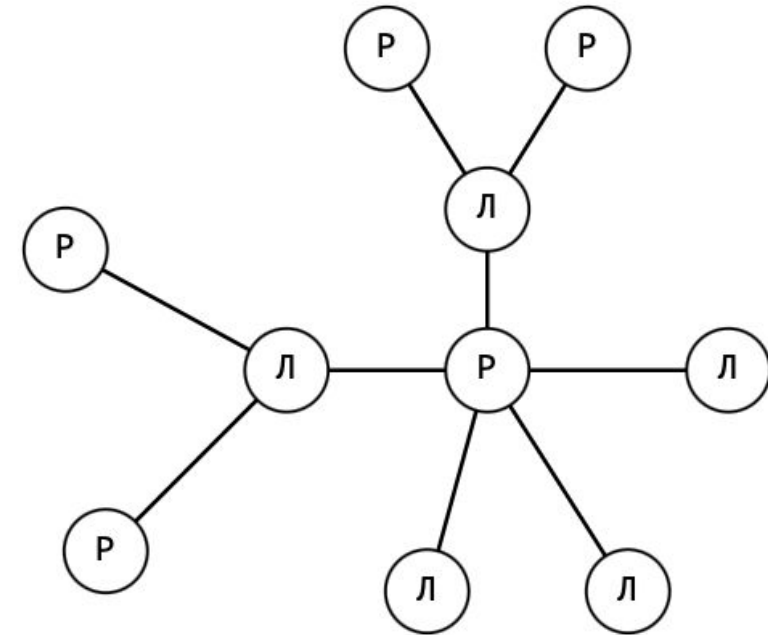
Гипотеза о
разметках на
графах. Программа
на Python

03

Задача про рыцарей и лжецов

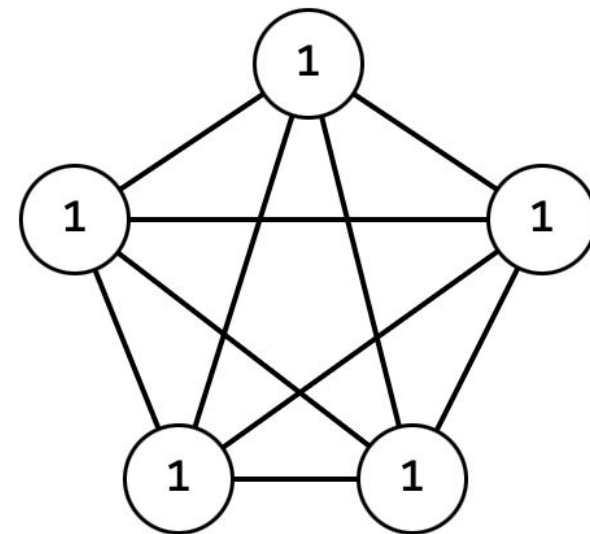
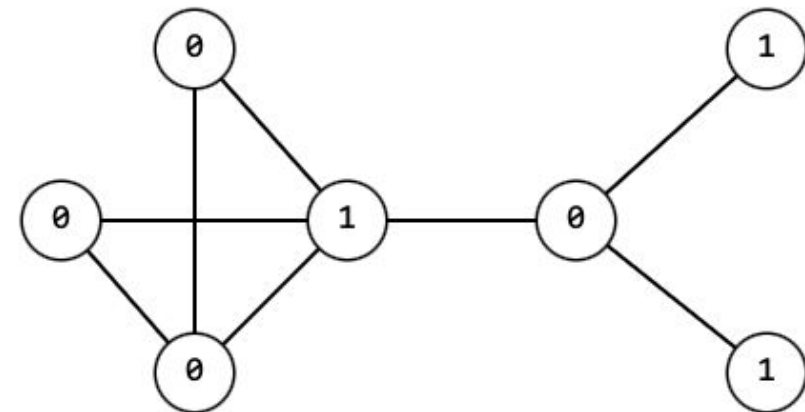
В вершины произвольного дерева расставили рыцарей и лжецов. Каждый из них заявил: “Среди моих соседей нечётное число лжецов”. Могло ли такое произойти на любом дереве? Лжецы всегда лгут, а рыцари всегда говорят правду.

Л – лжец
Р – рыцарь



Гармоническая разметка графа –

разметка вершин графа 0 и 1
таким образом, что в каждом
шаре радиуса 1 сумма меток
нечётна.



Рыцари и лжецы



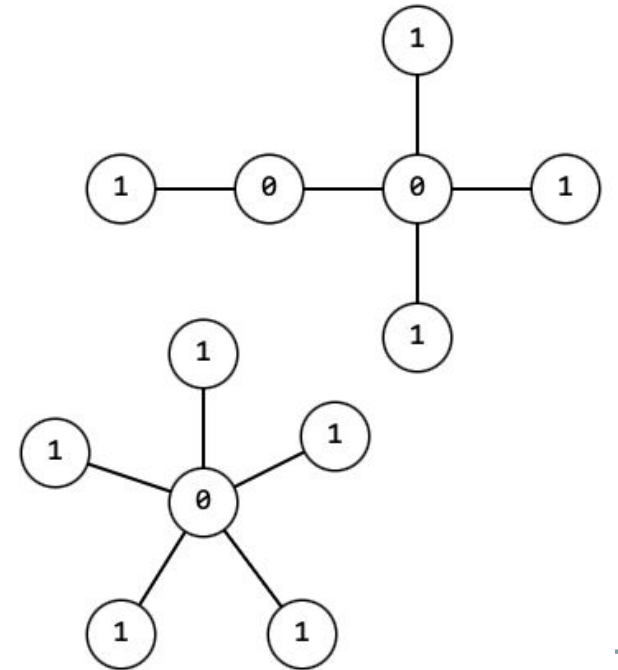
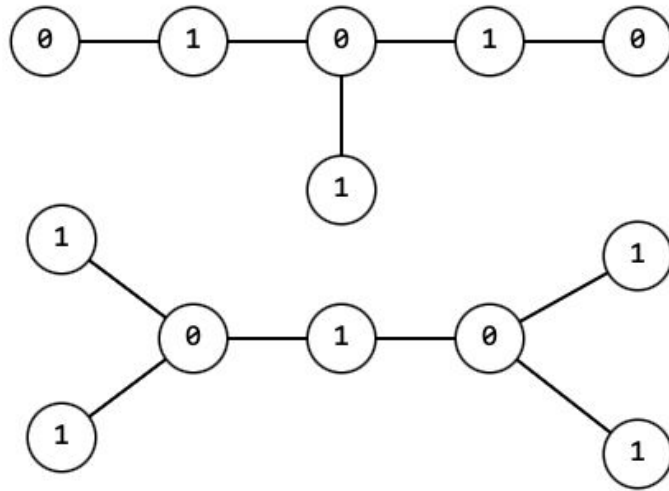
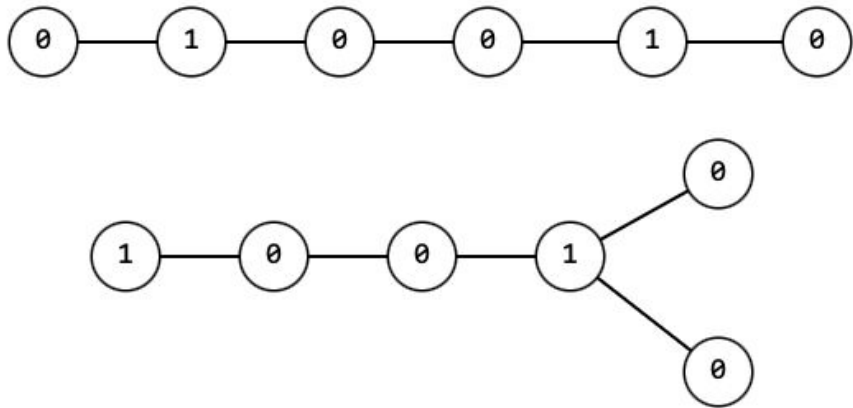
Гармоничные разметки

- Рыцари
- Лжецы
- Высказывание: "Среди моих соседей нечётное число лжецов"

- 0
- 1
- Условие: "В каждом шаре радиуса 1 сумма меток нечётна"

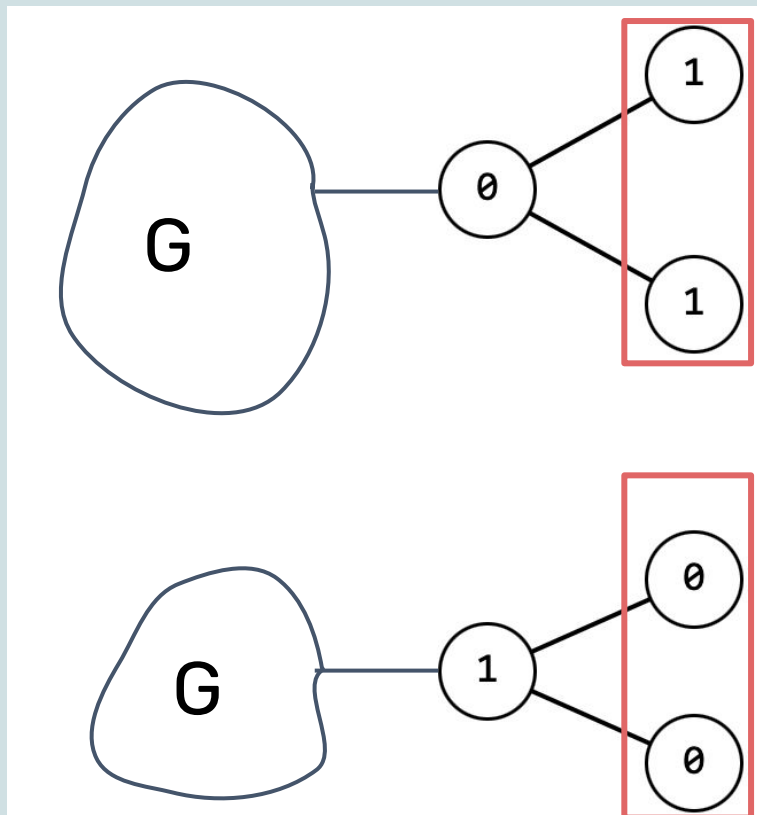
Теорема

Каждое дерево имеет как минимум одну гармоническую разметку

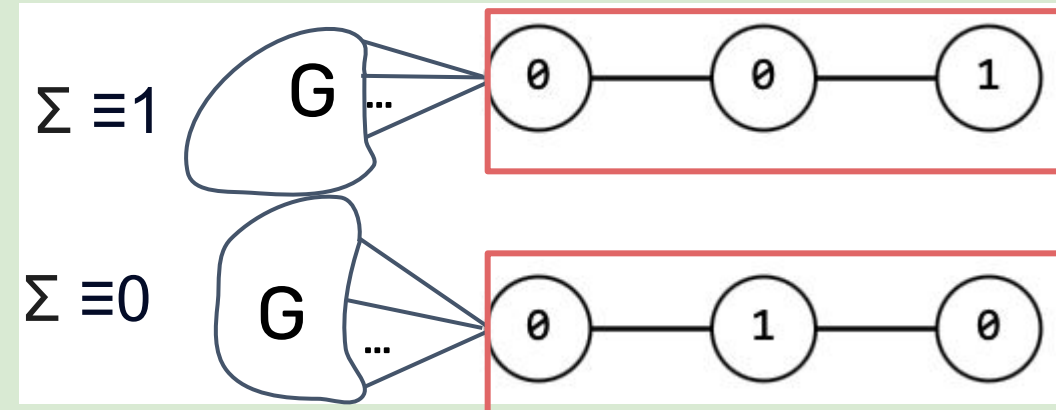


Редукции

1.) Редукция двух соседних висячих вершин

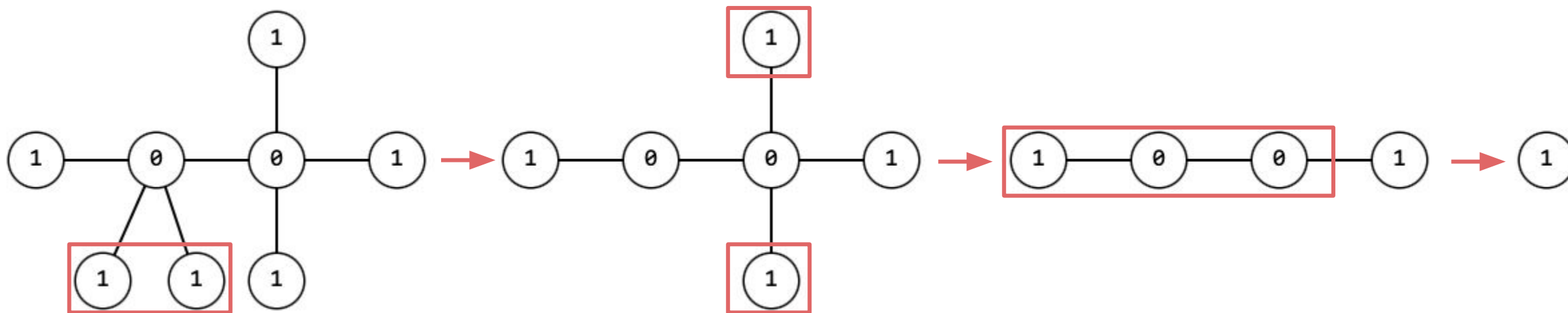


2.) Редукция висячей трёхвершинной цепи без ветвлений



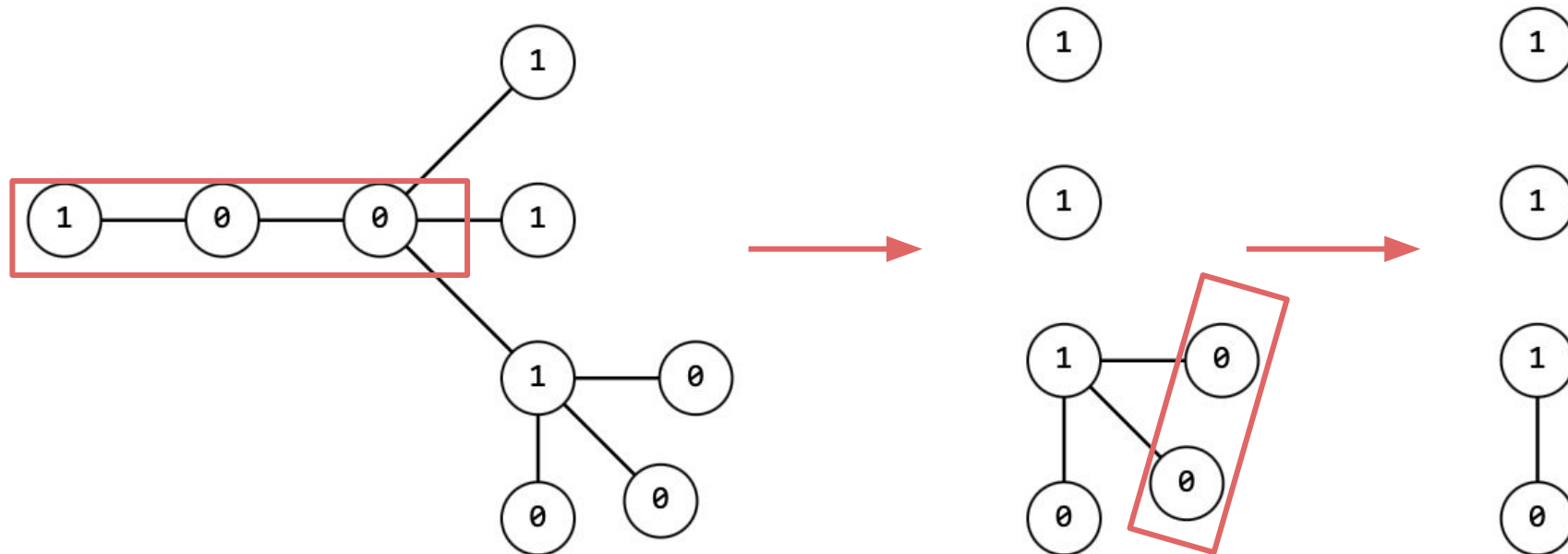
Лемма 1 (о редукциях)

Каждая из редукций 1 и 2 не меняет число гармонических разметок первоначального графа дерева.



Лемма 2 (о конечном числе редукций)

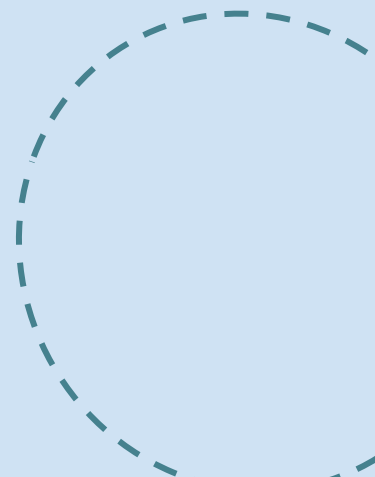
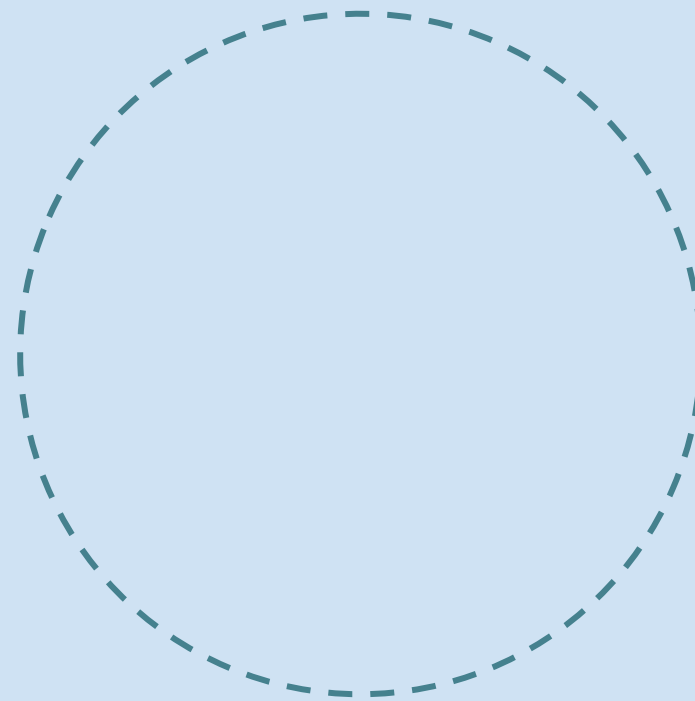
к любому дереву, если оно не изолированная вершина и не изолированное ребро, можно применить редукцию 1 или 2.

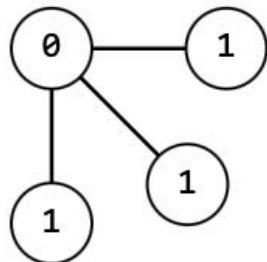
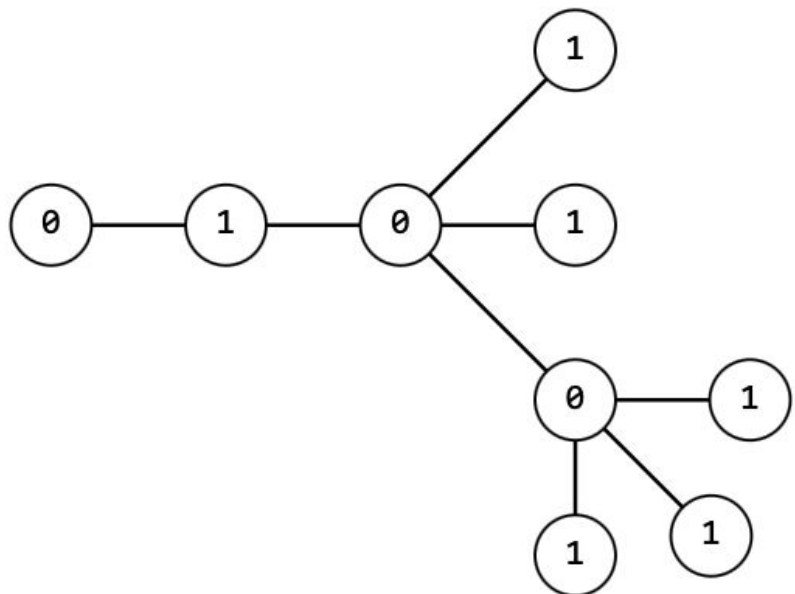
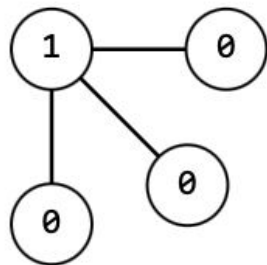
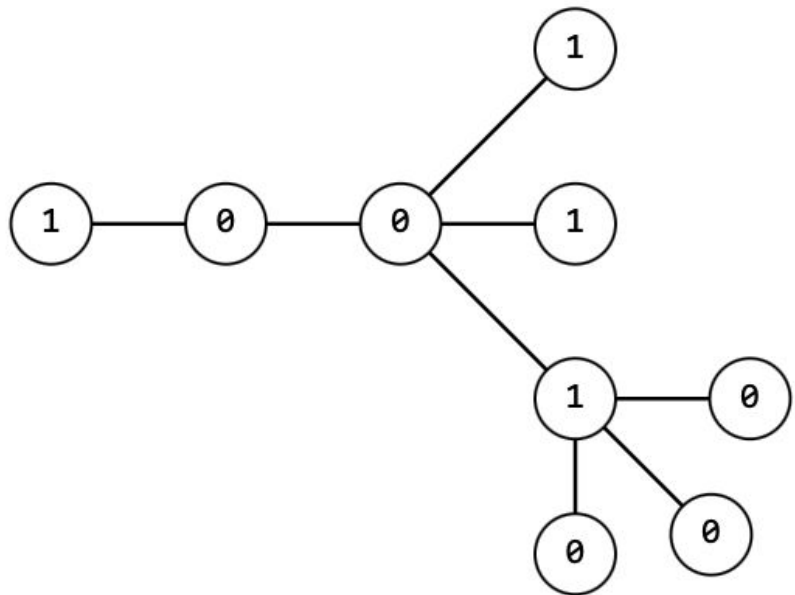




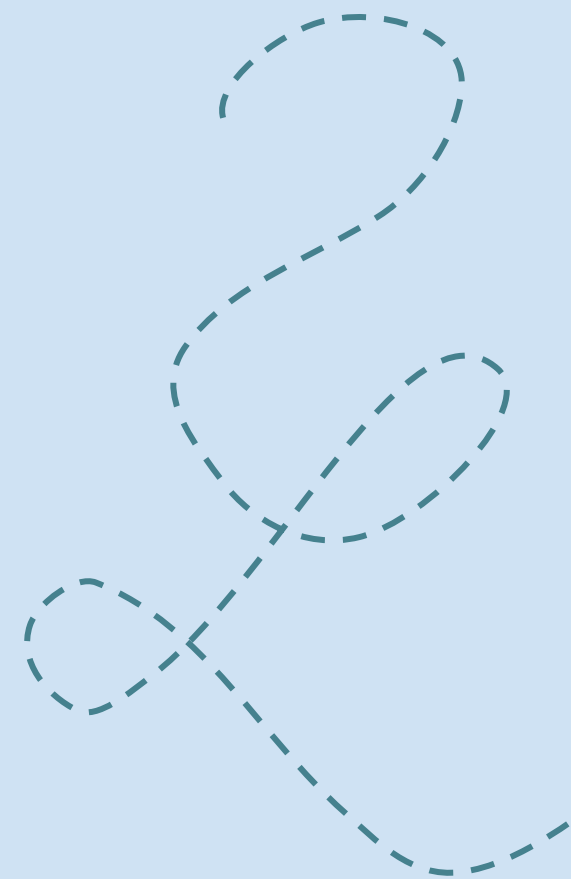
Следствие из лемм 1 и 2

Если в конце осталось r
одинокных рёбер, то
количество
гармонических разметок равно 2^r





Пример:





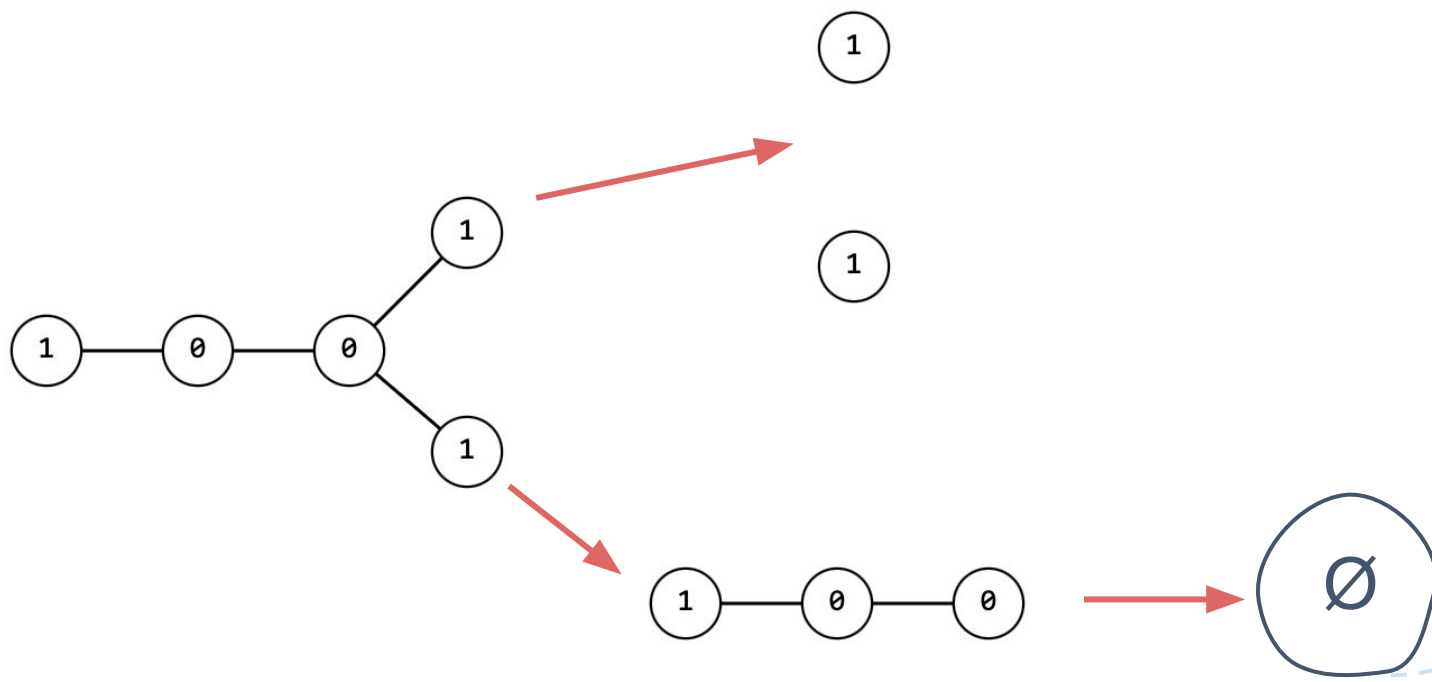
Вопрос для аудитории:

Зависит ли результат от порядка
применения редукций?



Вопрос для аудитории:

Зависит ли результат от порядка применения редукций?



пустой граф (0
изолированных рёбер
и вершин) имеет одну
гармоническую
разметку

Обобщение про графы:

Эксперименты с графами вплоть до $N = 5$ (где N – количество вершин в графе) показали, что для каждого из них гармоническая разметка существует.

Мы сформулировали гипотезу о том, что для каждого графа существует гармоническая разметка.

При проверке графов на наличие гармонических разметок, удалось выделить несколько замечаний:

Замечания

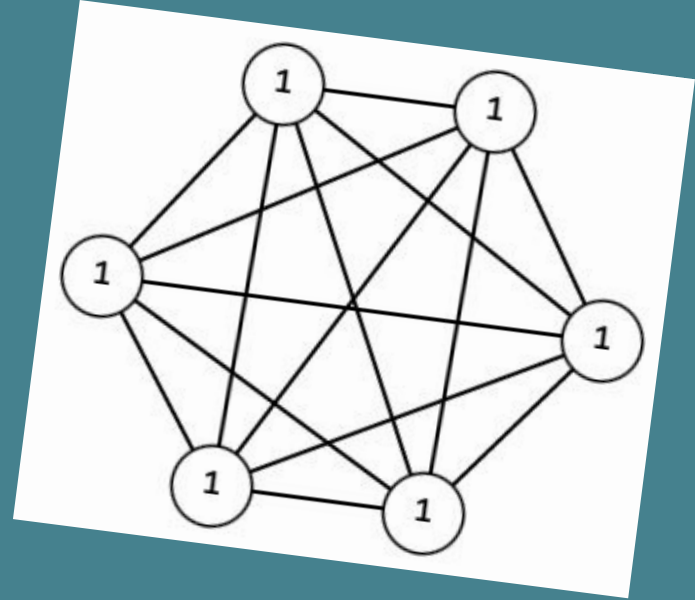
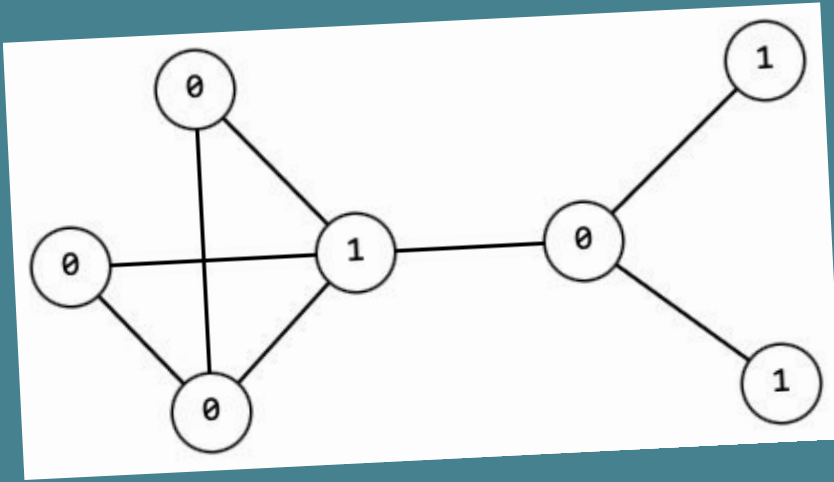
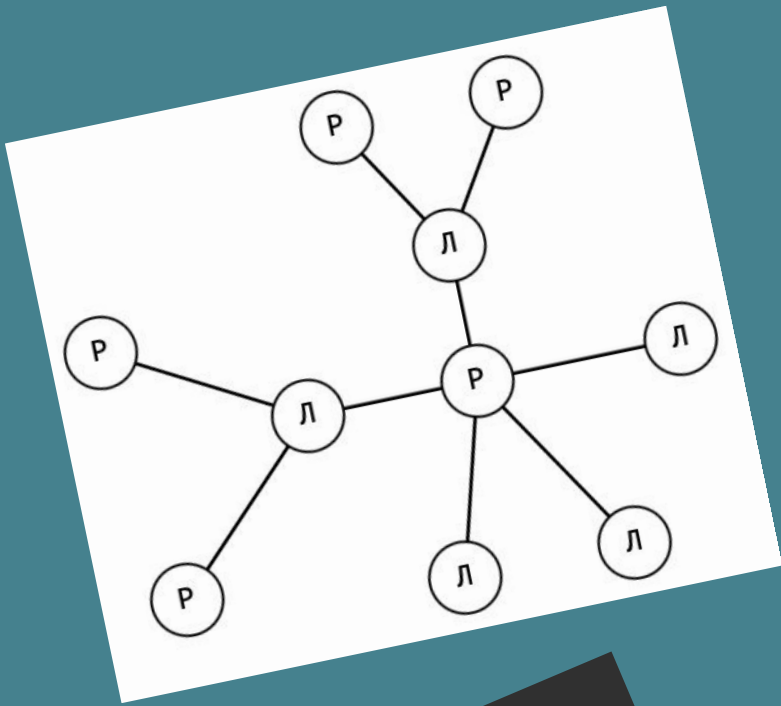
- 1) Любой граф с вершинами, каждая из которых четной степени, имеет хотя бы одну гармоническую разметку (сплошную из 1)
- 2) Полный граф на N вершинах можно гарантированно разметить гармонически

Продолжение исследования:

Основная задача дальнейшего исследования – дописать программу, которая рассчитана на поиск гармонической разметки для каждого N -вершинного графа (при отсутствии разметки для какого-либо графа гипотеза будет опровергнута)

*В перспективе при отсутствии контрпримеров на определенных порядках графов нужно доказать гипотезу.

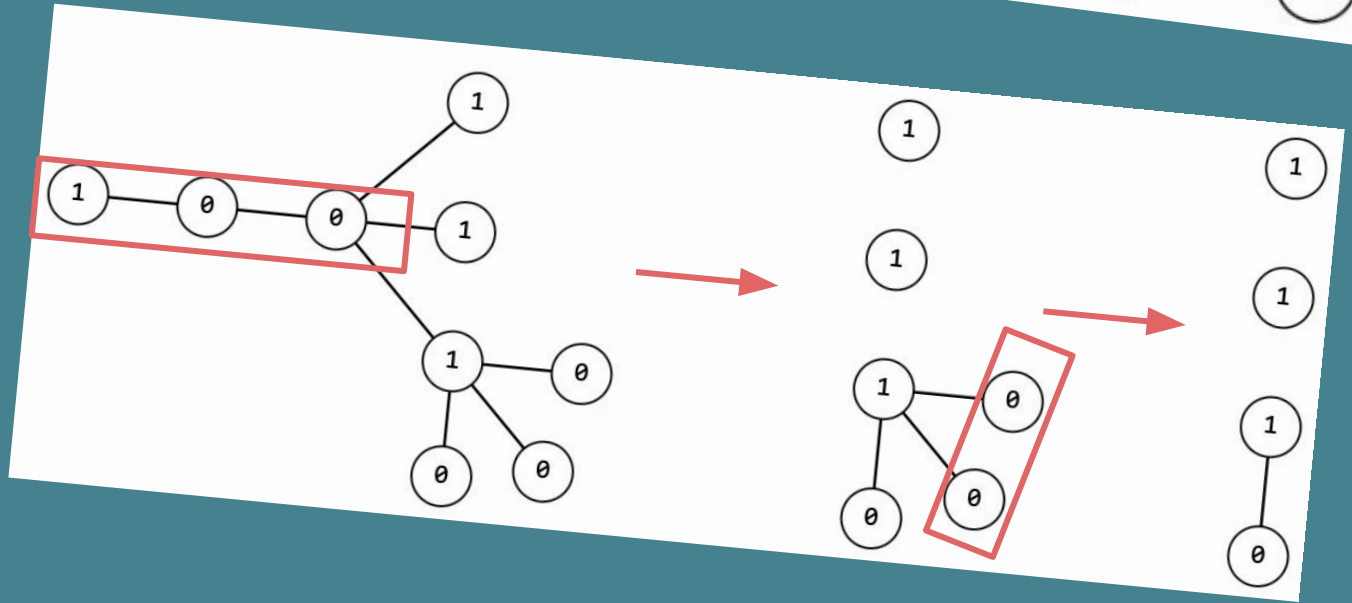
```
1 l = 6
2 r = l*(l-1)//2
3 arr = []
4 for n in range(2**r):
5     a = str(bin(n))[2:]
6     while len(a) < r:
7         a = '0' + a
8     m = []
9     for i in range(l):
10        m.append([])
11        for j in range(l):
12            m[-1].append(0)
13    arr.append(m)
14    c = 0
15    for i in range(l):
16        for j in range(i+1, l):
17            arr[-1][i][j] = int(a[c])
18            c += 1
19 for n in range(2**r):
20     with open('output 6.txt', 'a') as f:
21         f.write(str(n)+'_'+str(arr[n])+'\n')
22
```

```

1 l = 6
2 r = 1*(l-1)//2
3 arr = []
4 for n in range(2**r):
5     a = str(bin(n)[2:])
6     while len(a) < r:
7         a = '0' + a
8     m = []
9     for i in range(l):
10        m.append(0)
11        for j in range(l):
12            m[-1].append(0)
13        arr.append(m)
14        c = 0
15        for i in range(l):
16            for j in range(i+1, l):
17                arr[-1][i][j] = int(a[c])
18                c += 1
19        for n in range(2**r):
20            with open('output 6.txt', 'a') as f:
21                f.write(str(n) + ' ' + str(arr[n]) + '\n')
22

```



Спасибо за внимание!!!