Коллоквиум по дисциплине: Комбинаторика и Теория Графов
(Задания на создание программных модулей в системе “Wise Task Graph”)

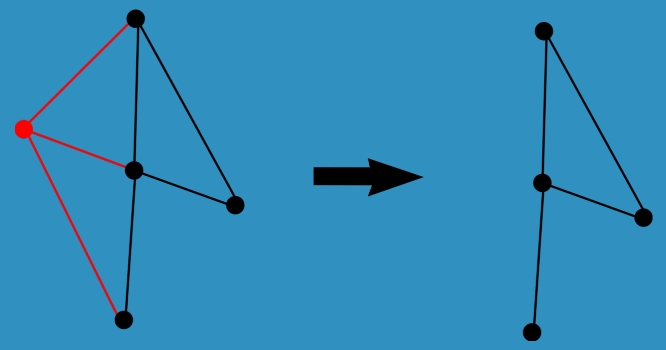
выполнил студент группы 3384
Пьянков Михаил

Задание

Проверка содержит ли граф индуцированный полный двудольный подграф K_{2,2}

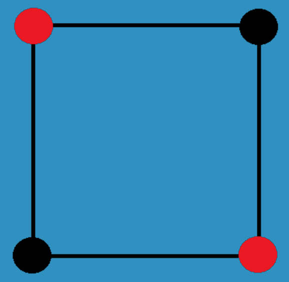
Терминология:

- индуцированный подграф это граф, образованный из подмножества вершин графа вместе со всеми рёбрами, соединяющими пары вершин из этого подмножества.



- полный двудольный граф — вид двудольного графа, у которого любая вершина первой доли соединена со всеми вершинами второй доли.

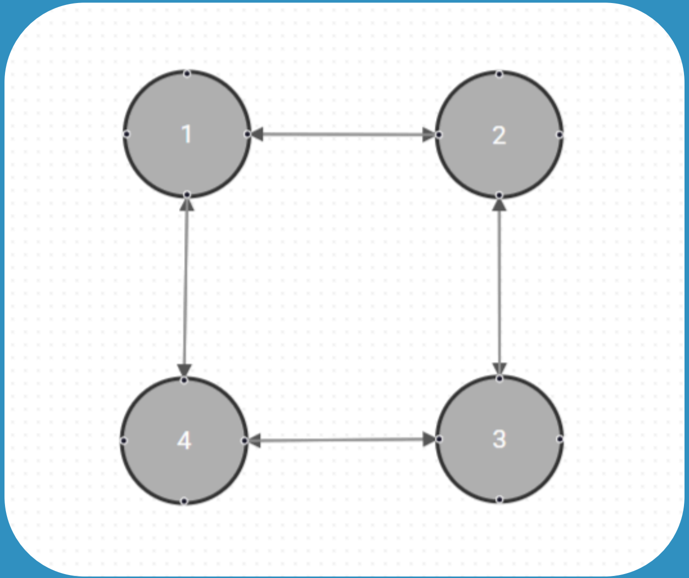
- граф K_{2,2}: представляется в виде двух множеств пар вершин V_1 = {u, v}, V_2 = {w, z}, при этом связей между элементами одного множества нет, и присутствуют все связи между элементами разных множеств



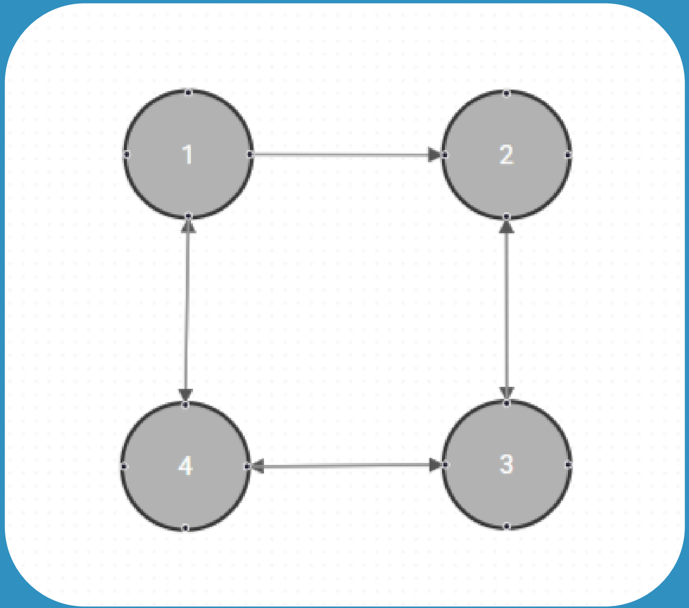
Задание

В задании не указано для какого графа осуществляется проверка: ориентированного или неориентированного.

Так как по условию искомый граф обладает условием полноты, то при переборе вариантов в ориентированном графе будем требовать наличие двунаправленных рёбер.



Подходящий вариант



Неподходящий вариант

Решение

Рассмотрим различные варианты решения данной задачи

- Наивное решение: перебрать все четвёрки вершин в исходном графе и проверить двудольность такого подграфа. Количество операций: C(n,4) * c, где c - количество операций для проверки, если нашли (a,b,c,d), то надо проверить C(4,2) вариантов разделения на {(u,v), (m,n)}. То есть: (u,m) in E, (u,n) in E, (v,m) in E, (v,n) in E, (u,v) not in E, (m,n) not in E. Таким образом c = C(4,2) * 6.

  Итоговое число операций 9n*(n-1)*(n-2)*(n-3), O(n^4).

- Использование возведения матрицы смежности в 4-ую степень, проверка диагональных элементов и в случае не нулевого значения запуск из них DFS глубиной 4. Возведение матрицы смежности в 4-ую степень это 3*n^3 операций. Проверка диагональных элементов: n*1 операций. DFS с глубиной 4: (n-1)*(n-2)*(n-3)*1 операций.

  Итоговое число операций: 3*n^3 + n*(n-1)*(n-2)*(n-3), O(n^4).

Решение

- Перебор пар несмежных вершин (u, w), поиск пар несмежных вершин (a,b) в пересечении множеств соседей N(u) и N(w). При таком подходе не требуется проверка на двудольность. Перебор несмежных пар: C(n,2)*1. Построение пересечения множеств соседей: n. Поиск несмежных пар в пересечении множеств соседей C(n-2, 2)*1.
  Итоговое количество операций: n*(n-1)*(n-2)*(n-3)/4, O(n^4).

В каждом из вариантов также необходимо проверить все 4 вершины найденного подграфа на отсутствие петель: при выборе очередной вершины v проверяем что нет ребра (v,v). Такая проверка производится за 1 операцию, поэтому не влияет на сравнение методов.

Последний вариант решения предлагает наибольшую эффективность вычислительных затрат, поэтому он и будет реализован.

Решение

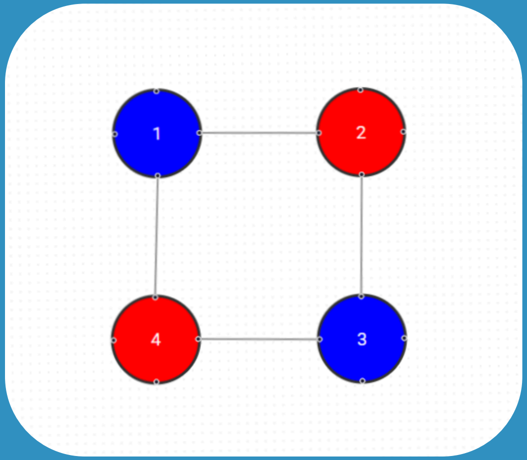
Псевдокод:

```
// Инициализация списка смежности
adj = НОВЫЙ_СЛОВАРЬ()
for v in V:
    adj[vertex] = Ø

// Заполнение списка смежности
for e=(u, v) in E:
    adj[u].add(v)
    if граф неориентированный:
        adj[v].add(u)

// Поиск индуцированного полного двудольного подграфа K_{2,2}
for u in V:
    // Пропуск, если есть петля у u
    if u ∈ adj[u]:
        CONTINUE
    for v in V, v > u: // Сравнивается номер вершины, чтобы не повторяться
        // Пропуск, если u и v смежны или есть петля у v
        if v ∈ adj[u] или v ∈ adj[v] или (граф ориентированный и u ∈ adj[v]):
            CONTINUE

        // Нахождение общих соседей u и v
        common = adj[u] ∩ adj[v]
        if |common| < 2:
            CONTINUE

        if граф ориентированный:
        {
            // Проверка двунаправленности рёбер для общих соседей и удаление неподходящих вариантов
            common = common \ {w: | v ∉ adj[w] или u ∉ adj[w]}
        }

        // Перебор пар общих соседей
        for w in common:
            // Пропуск, если есть петля у w
            if w ∈ adj[w]:
                CONTINUE
            for z in common, z > w:
                // Если w и z не смежны и у z нет петли, то такой вариант подходит
                if z ∉ adj[w] и не(z ∈ adj[z]) и не(граф ориентированный и w ∈ adj[z]):
                    EXIST 1  // Найден цикл u-w-v-z

EXIST 0
```

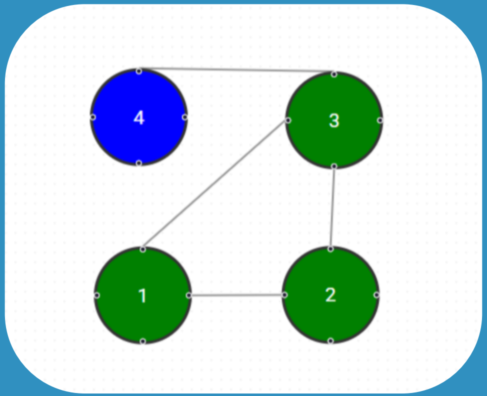
Обоснование корректности:

- Инвариант: в цикле для каждой пары вершин (u, v) сохраняются следующие свойства: u и v принадлежат одной доле V_1 искомого K_{2,2}, между u и v нет связей; во вложенном цикле пара вершин (w, z) из множества общих соседей u и v образует вторую долю V_2 графа K_{2,2} и между ними также нет связей.
- Корректность: Пусть существует индуцированный полный K_{2,2} = (V_1, V_2), где V_1 = (u, v), V_2 = (w,z), тогда ребра (u, w), (u, z), (v, w), (v, z) существуют так как рассматриваются общие соседи u и v, рёбер (u, v) и (w, z) нет, так как проверялись условия смежности, что соответствует определению K_{2,2}.
- Полнота: циклом перебираются все пары вершин (u,v) из V - множества всех вершин, для вершин u и v перебираются все пары из их общих соседей.

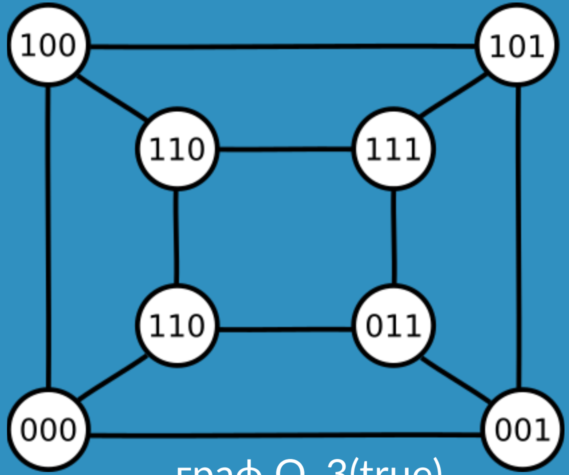
Тестирование (неориентированный граф)



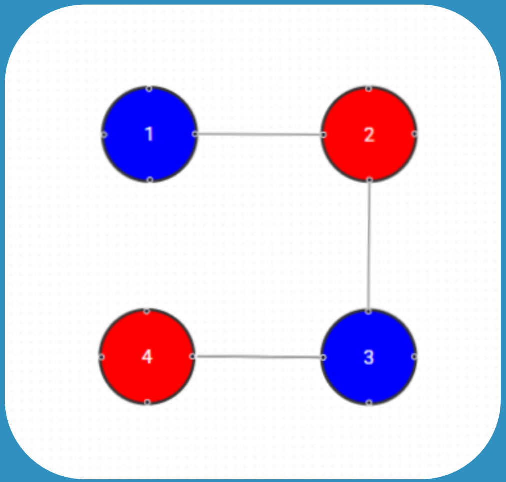
граф K_{2,2}(true)



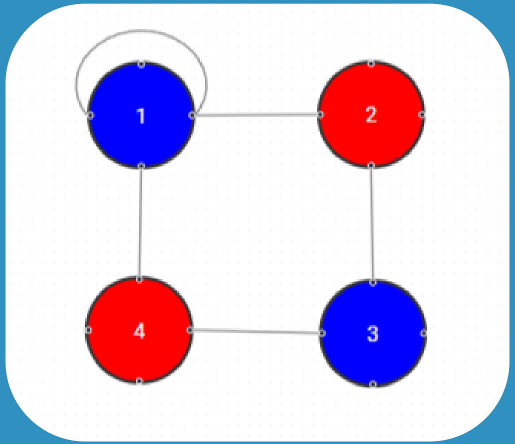
граф - треугольник с доп. вершиной(false)



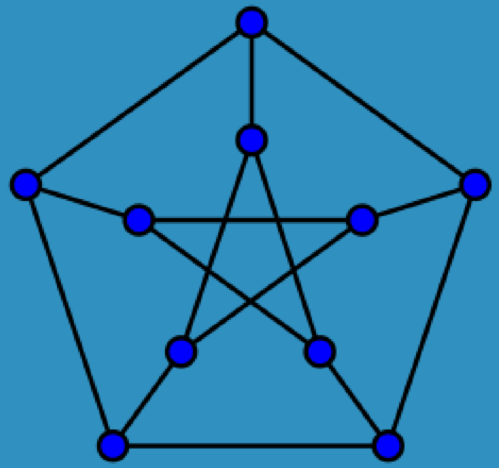
граф Q_3(true)



граф P_4(false)

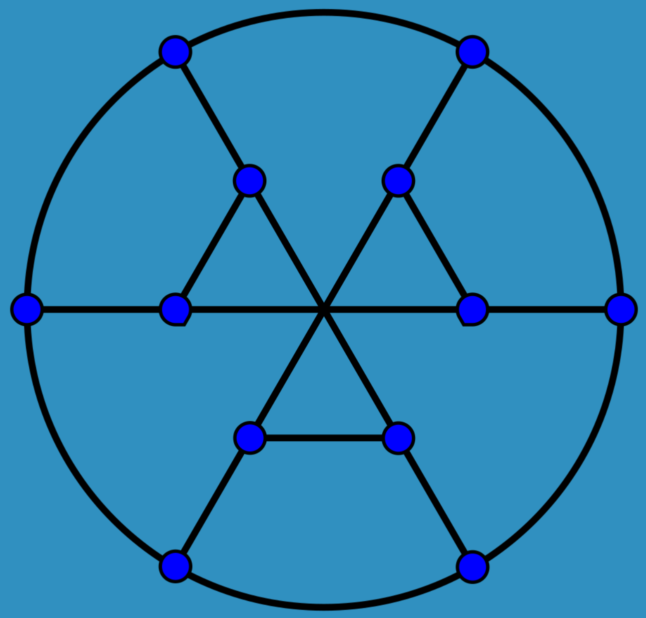


граф K_{2,2} с петлёй (false)

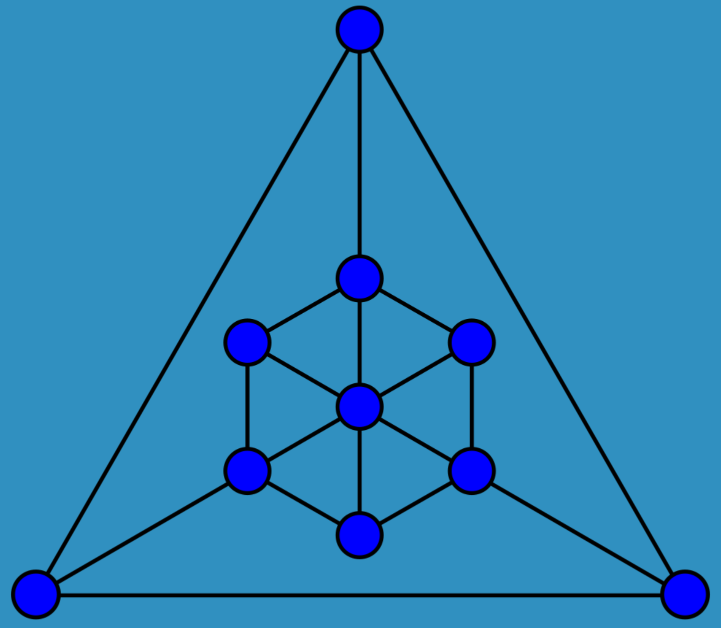


граф Петерсена(false)

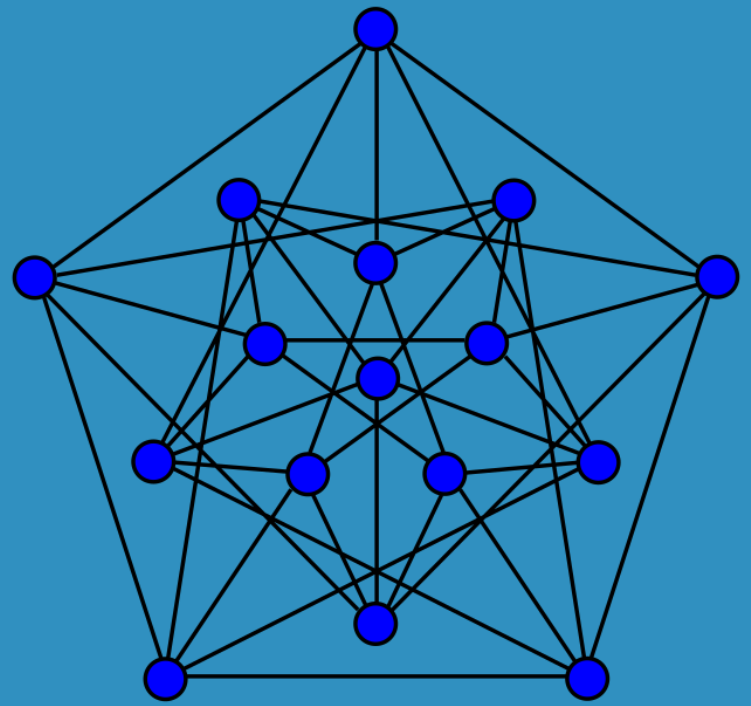
Тестирование (неориентированный граф)



граф Франклина(true)

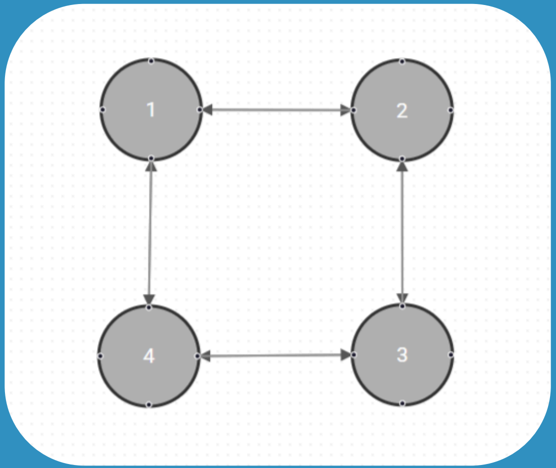


граф Голомба(false)

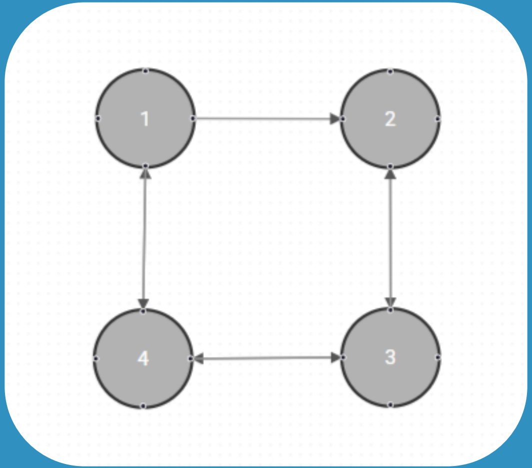


граф Клири(true)

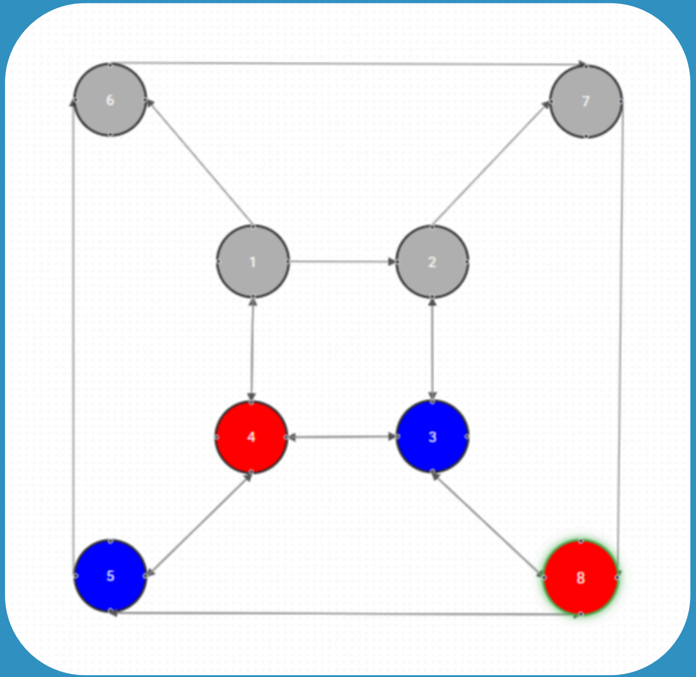
Тестирование (ориентированный граф)



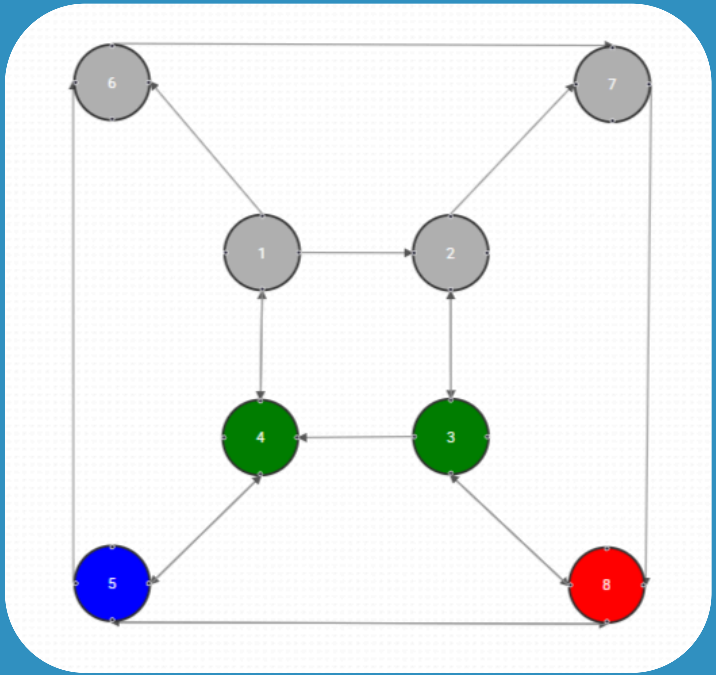
граф K_{2,2}(true):



граф K_{2,2}(false):



граф Q_3(true):



граф Q_3(true):

Спасибо за внимание

Ссылка на код: https://clck.ru/3MLvPw