4 задача: Двудольный граф

1 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

Дан невзвешенный неориентированный граф. Определить, является ли он двудольным. Требуемая сложность O(V+E).

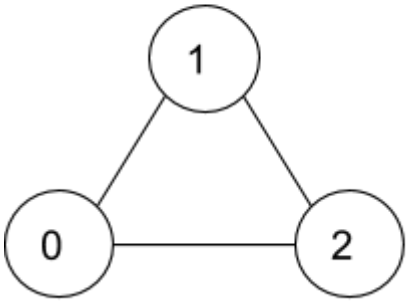
Ввод: v:кол-во вершин(макс. 50000), n:кол-во ребер(макс. 200000), n пар реберных вершин.

Вывод: YES если граф является двудольным, NO - если не является.

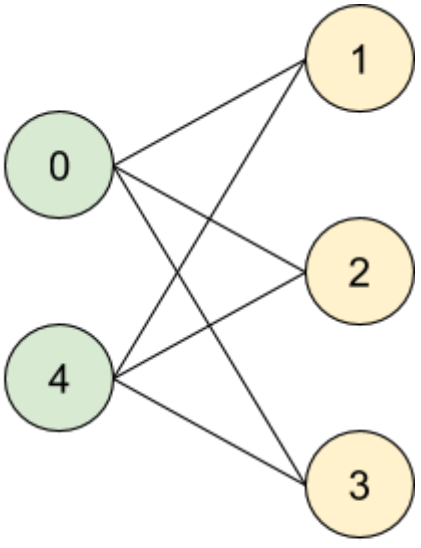
Пример:

in	out
3 3 0 1 1 2 0 2	NO
5 6 0 1 0 2 0 3 1 4 2 4 3 4	YES

Иллюстрация примера:



Разделение на доли не возможно



2. Описание алгоритма

Запускаем BFS от любой вершины в каждой компоненте связности. Красим все дочерние вершины в цвет, противоположный цвету текущей вершины (цвет определяет долю вершины). Если в процессе покраски одна из дочерних вершин уже имеет цвет, совпадающий с цветом текущей вершины (то есть есть ребро в вершину в той же доле), то

граф не является двудольным и алгоритм завершается. Если же таких случаев не было, то граф двудольный

3. Доказательство корректности

Докажем в обе стороны следующее утверждение:

Алгоритм говорит, что граф не двудольный ⇔ граф действительно не двудольный

$(\Rightarrow)$

Пусть у нас уже есть корректное разделение на доли некоторого подграфа. Если при обработке v мы нашли ребро, которое идёт в вершину w той же доли, то двудольность нарушается. Докажем отсутствие такого альтернативного разбиения на доли, в котором v и w лежат в разных долях. Предположим существование такого разделения.

Переместим v в другую долю, тогда придётся также переместить все её соседние вершины (кроме w) в другую долю, и так далее. В итоге переместить придётся соседей w (мы до них дойдём потому что есть цикл между v и w через начальную вершину обхода), а потом и саму вершину w. В итоге w и v опять окажутся в одной доле, противоречие

$(\Leftarrow)$

BFS обходит каждое ребро, следовательно он найдёт ребро, нарушающее двудольность

4. Время работы и дополнительная память

$T = O(V + E)$

$M = O(V)$

5. Доказательство времени работы и дополнительной памяти

а) Время работы:

Алгоритм состоит из одного запуска BFS для каждой компоненты связности, суммарное время работы составляет $T = O(V + E)$

б) Дополнительная память:

Размер контейнеров для: очереди, буфера для следующих вершин, цвета - не превышает V, значит $M = O(V)$