МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИДЗ

по разработке и анализу алгоритмов

“Последовательное исключение степеней”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С. М.

(подпись, дата)

Группа АС-21-1

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Гаев Л. В.

(подпись, дата)

Липецк 2022 г.

Содержание:

2. Задание

3. Решение

4. Трудоемкости

5. Доказательство NP-полной задачи

2. Задание:

ТГ 47.

Заданы граф G = (V, E) и последовательность <d1, d2, ..., d|v|> неотрицательных целых чисел, не превы­шающих |V| - 1.

ВОПРОС. Существует ли взаимно однозначная функция f: V → {1, 2, ..., |V|} со следующим свойством для всех i, 1 ≤ i ≤ |V|: если f(v) = i, то существует в точности di вершин u, таких, что f(u) > i и {u, v} ∈ Е?

Дано:

G = (V, E)

d = <d1, d2, ..., d|V|>, di <= |V| - 1

1 <= i <= |V|

f(v) = i

f(u) > i

{u, v} ∈ E

Вопрос:

∃ f: V -> {1, 2, ..., |V|} ?

Входной поток:

N, Матрица смежности N^2 элементов

d, массив d[V], где V – число вершин, полученных из матрицы смежности

Отсюда мы получим множество всех вершин A=V, число их степеней d

Выходной поток:

Вывод bool-значения [true – функция существует, false – функция не существует]

3. Решение:

Алгоритм выбирает множества A, руководствуясь следующим правилом: на каждом этапе выбирается множество, покрывающее максимальное число ещё не покрытых элементов, т.е.

bool f(A, d)

T ∈ P // T – хранит покрытые элементы

if (A[i](d[i]) == ∅) // Если в графе нет вершины степени d[i]

return false // то функции не существует

else

while V != P // V - контейнер для непокрытых элементов A[i]

select A[i](d[i]) ∈ A(d) // степень, которая покрывает максимальное число элементов в V

T += A[i](d[i])

V \= {A[i]}

end while

return true

Трудоемкость алгоритма в худшем случае:

f(A, d)

T ∈ P

if (A[i](d[i]) == ∅)

return false

else

while V != P // ((2+7n) + 1) \* n = 3n + 7n^2

select A[i](d[i]) ∈ A(d) // 2+7n

T += A[i](d[i]) // 4

V \= {A[i]} // 3

end while

return true

// 13n^2 = O(n^2)

Трудоемкость алгоритма в лучшем случае:

f(A, d)

T ∈ P

if (A[i](d[i]) == ∅) // 3

return false

else

while V != P

select A[i](d[i]) ∈ A(d)

T += A[i](d[i])

V \= {A[i]}

end while

return true

// O(n)

Трудоемкость алгоритма в среднем случае == трудоемкости в худшем случае:

O(n^2)

Доказательство того, что задача является NP:

Известно, что задача о точном покрытии 3-множествами является NP.

Можно доказать, что сводится к задаче о точном покрытии 3-множествами:

Смысл задачи в том, что есть множества C’∈ C ∈ X. Можем ли мы найти подмножество C’ множества C, где каждый элемент X встречается ровно в одном элементе C’?

Т.е если X {1,2,3,4,5,6}, а С {{1,2,3},{2,3,4},{1,2,5},{2,5,6}, {1,5,6}} и C’ {{2,3,4},{1,5,6}} – подходит.  
Если C {{1,2,3},{2,4,5},{2,5,6}}, то что бы мы ни выбрали в C’ условие не выполнится, т.к. везде присутствуют ‘2’.

Если в нашей задаче принять P за X, V за C, то A – это C’ из приведенной выше задачи.

Т.к. вершины зависят от значений множества V, а подмножество V, в свою очередь, зависит от значений множества P.

Например, при P = {1, 2, 3, 4, 5, 6, 7, 8, 9} и разбиении его на V = {{1, 3, 4}, {2,5,7},{6,8,9}, {1,2,5}, {3, 4, 9}, {6,7,8}} вершины {{1, 3, 4}, {2, 5, 7, {6, 8, 9}} подойдут. В другом же случае, если значения множества V {{1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {3, 6, 9}, {5, 6, 7}, {4, 5, 6}}, то условие задачи также не выполнится.

Вывод: попробовал решить NPC задачу, посчитал её трудоемкость, доказал её принадлежность к классу NP.