ТГ 47 ПОСЛЕДОВАТЕЛЬНОЕ ИСКЛЮЧЕНИЕ СТЕПЕНЕЙ

Дано:

G = (V, E)

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

Вопрос:

∃ f: V -> {1, 2, ..., |V|} ?, 1 <= i <= |V|, f(v) = i, f(u) > i, {U, V} ∈ E

Входной поток:

Множество всех вершин A, число степеней d.

Выходной поток:

Вывод возможного множества S

Решение:

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

```
f(A, d)

T ∈ P // T – хранит покрытые элементы

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

      return // 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 A
```

Трудоемкость алгоритма в худшем случае:

```
f(A, d)
T ∈ P
if (A[i](d[i]) == ∅)
      return
else
      while V != P // ((2+4+3) + 1) * n
            select A[i](d[i]) ∈ A(d) // 2
            T += A[i](d[i]) // 4
            V \= {A[i]} // 3
      end while
return A
// 10n = O(n)
```

Трудоемкость алгоритма в лучшем случае:

```
f(A, d)
T ∈ P
if (A[i](d[i]) == ∅) // 3
      return
else
      while V != P
            select A[i](d[i]) ∈ A(d)
            T += A[i](d[i])
            V \= {A[i]}
      end while
return A
// 3
```

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

```
O(n)
```

Доказательство того, что задача является NP-Complete:

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

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 C {{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.