Гавриленко Арсений *777*

Домашняя работа 3

Задача 1.

1. Воспользуемся материалом прошлого задания (строили Мега МТ для проверки замкнутости Р относитнльно итерации). ПОдадим все сертификаты (а они существуют тк языки NP) как один Мега Сертификат-тогда мы свели задачу к предыдущией(про Р). Значит класс NP тоже замнкут относительно итерации.

2.

$$P \subseteq co - NP?$$

$$if L \in P => \overline{L} \in P => \overline{L} \in NP => L \subseteq co - NP$$

Задача 2.

1. Сертификат - правильный набор. Его подстановка занимает линейное время. Тогда 3КНФ - лежат в NP.

2. Язык графов - VCOVER, имеющий вершинное покрытие заданной мощности. Построим МТ - на вход пара (G, k), сертификат (понятия не имею как его найти) - то самое вершинное покрытие , может быть записан как матрица смежности или же ничего - если такого покрытия нет. Тогд чтобы принять весь этот набор, сначала надо проверить что |y| = k (покрытие реально из к вершин). И потом проверить- что это покрытие - то есть что все ребра трансцинденты хотя бы одной вершине из покрытия - пройдемся по всем ребрам × всем вершинам(как максимум) - за полином. Тогда исходный язык в NP.

3. Сертификат - список вершин w. Пусть Граф представлен матрице смежности. ТОгда проверка, что путь реально эйлеров займет O(|w|*|E|*|E|) - то есть полином.

4. Мт на вход - описание графа(матрица), далее Белман-Форд, который либо находит какой-то цикл отрицательной длины, либо говорит что его , очевидно что такая МТ разрешает язык. Алгоритм - полиномиальный $=> L \in P => L \in NP$

5.

6. PLANARITY - записи всех планарных графов. Для связного графа можно проверить его планарность, посчитам эйлерову характкеричтику: V-E+F, F - число граней. Если входной граф не связный- проверим планарность каждой его компоненты, перед этим с помощью BFS найдем их все. F - колво граней можно найти за $O(|V|^2)$ - существует алгоритм, см EMAX. Таким образом, язык явлесят даже P, тогда очевидно он NP.

Задача 3.

1. Докажем, что $L = \overline{T}AUT$ лежит в $\mathcal{NP}$. Данный язык состоит из необщезначимых формул, значит есть хотя бы один набор на котором она не истина. Тогда понятно, что если на вход некой MT мы подадим нашу формулу из сертификат на котором она не истина - за полином получим проверку. Получили, что L $L = \overline{T}AUT \subseteq NP => TAUT \in co - NP$

2.

3. FACTORING - язык натуральных троек (a, b, c) таких, что a имеет простой делитель из $[b, c]$. Построим $L = \overline{F}ACTORING$, тогда язык L состоит из троек, для которых a не имеет простых делителей из $[b, c]$. Тут два случая, либо на этом отрезке в принципе нет простых чисел, либо там нет делителей нашего числа a. Построим MT, которая принимает (a,b,c,y), y -некоторый сертификат. Теперь первый случай проверим решетом Эратосфена - $(O(nlog(logn))$, если на отрезке не нашлось простых - машина прнимает данный вход. Дальше нам нужно изнать 1. является ли y делителем a и 2. лежит ли y между b c 3. простое ли оно . начнем с 1.

- Проверим является ли y делителем a. MOжно просто запустить gcd(a,y) и если он равен y - y делитель. асимптотика O(log(min(a,y))) - то есть полином
- Теперь проверим, что простое - сущестувут известный алгоритм O(log y). подразумеваем что машина M умеет читать десятичные числа.

Если на отрезкеРассмотрим MT $A(x, y)$, в которой $x = (a, b, c)$ которая в первую очередь, пользуясь решетом Эратосфена, проверяет есть ли вообще на отрезке $[b, c]$ простые числа, если таких нет, алгоритм выдает

1. Затем проверим делимость числа a на y, это можно сделать просто переборем все натуральные чисел z из отрезка $[1, a]$ и умноженим их на y, если не нашлось такого числа, что $a = yz$, то выдаем 1, в противном случае проверяем лежит ли число y в $[b, c]$, это делается просто побитовым сравнениями, начиная со сташего бита, эсло этот пункт . Таким образом, данная МТ работает за полиномиальное время. Сертификатом в данном случае будет простое число из отрезка $[b, c]$, если оно существует и, к примеру, 0, если такого числа нет. Легко поянть, что данная МТ полиномиальна по входу и $|y| = O(|x|^c)$, значит данный язык лежит в $co - \mathcal{NP}$

4. Чтобы доказать что в графе есть клика - нужно сначал выбрать 2019 вершин - $C_n^k == O(n^{2019})$. Проверить - пройтись по нужным стобцам и строкам в матрцие смежности - $p(2019) = const$. Итого $L \in P => L \in NP$

5. аналогисно 2.6

Задача 4. Нужно проверить что $L \subseteq co - NP$, то есть, что $\overline{L} \in NP$. Язык $\overline{L}$ - задача проверке необщезначинмости предиката. Тогда если предьявим такой x, при котором предикат не общезначим(это вычислятся за полином от входа - по условию), значит задча лежит в NP. То есть мы доказали, что $\overline{L} \in NP$, значит доказали исходное.

Задача 5. n = 100091237, n-1 = 100091236

$$a = 7; 100091236 = 2^2 * 7 * 3574687$$

$$n_1 = 3574687, n_1 - 1 = 3574686$$

$$a = 2; 3574686 = 2 \cdot 3 \cdot 233 \cdot 2557$$

$$n_{21} = 233, n_{21} - 1 = 232$$

$$n_{22} = 2557, n_{22} - 1 = 2556$$

$$a = 2; 232 = 2^3 * 29; a = 2; 2556 = 2^2 * 3^2 \cdot 71$$

$$a = 2; n_{31} = 29; n_{31} - 1 = 28 = 2^2 * 7$$

$$a = 2; n_{32} = 71; n_{32} - 1 = 70 = 2 * 3 * 5$$

Теперь построим рекуренту и асимптотику. Минимальный делитель 2, поэтому высота дерева не более $log_2(p) = logp$. Можно реализовать наивную факторизацию за $O(\sqrt{(}p))$. Тогда :

$$T(p) = O(\sqrt{(}p)) + \Sigma_{k=1}^{i} T(\frac{p}{p_k})$$

Оценим сумму сверху так , возьмем максимальную цену операции, умножим на высоту дерева и умножим на максимальное колов обьектов для одного этажа(= колво листьев)

$$\Sigma_{k=1}^{i} T(\frac{p}{p_k}) \leq O(log(p)) * log(p) * log(p) = O(log^3(p) => \in P$$

Задача 6. По определению NP - длина сертификата y : $y \leq |x|^c$,x - описание условия/алгоритма. Но так как у нас унарный алфавит, то можно перебрать все варианты сертификата за полином. Тогда любой NP язык превщарается в язык P(ведь сертификат не надо искать , просто переберем все сертификаты линейно. так как по умолчанию $P \subseteq NP$, то получили P = NP(где получить 1кк USD?).

Задача 7.

1. Проверка того, что 2 графа изоморфны.

Назовем E^p — матрицу смежности графа, полученного перестановской вершин p в исходном графе с матрицей смежности E.

С учетом этого,

$$L = \{(G_1(V_1, E_1), G_2(V_2, E_2) \mid |V_1| = |V_2|, \exists p : E_1^p = E_2\}$$

Докажем, что данный язык лежит в $\mathcal{NP}$:

1. Сертификат y — перестановка вершин в графе G_1. Поскольку графы задаются матрицей смежности, длина записи составляет cn^2. Длина записи перестановки p не превышает $n \Rightarrow |y| \leq poly(|x|)$.

2. Верификатор R составляет матрицу E_1^p и поэлементно сравнивает ее с E_2. Время, затрачиваемое на составление новой матрицы (не более, чем $2n$ раз поменять столбцы/строки местами) — $O(n^2)$. Поэлементная проверка также занимает $O(n^2)$.

   Итак,

   $$|R(x, y)| = O(n^2) \leq poly(|x|)$$

Таким образом, мы показали, что язык L лежит в $\mathcal{NP}$.

2.

3.