Токмаков Александр, группа БПМИ165
Домашняя работа 2

№1

Нельзя утверждать, что $SAT \notin P$ только потому, что приведён экспоненциальный алгоритм решения. Возможно, существует полиномиальный алгоритм (нужно доказать, что не существует такого алгоритма).

№2

$SAT \leq_p \overline{TAUT}$ (если отрицание формулы выполнимо, то она не тавтология и наоборот) и $SAT \in NPC \Rightarrow \overline{TAUT} \in NPC \subset NP \Rightarrow TAUT \in coNP$. Более того, $TAUT \in coNPC$ т.к. если некоторый язык $L \in coNP$, то $\overline{L} \in NP$ и тогда $\overline{L} \leq_p \overline{TAUT} \Rightarrow L \leq_p TAUT$ т.е. любой $coNP$ язык сводится к $TAUT$.

Пусть $3SAT \leq_p TAUT$ и $TAUT \leq_p 3SAT$. Возьмём некоторый язык $L \in NP$:
$3SAT \in NPC \Rightarrow L \leq_p 3SAT \leq_p TAUT \Rightarrow L \in coNP \Rightarrow NP \subset coNP$.
Аналогично для некоторого языка $L \in coNP$:
$TAUT \in coNPC \Rightarrow L \leq_p TAUT \leq_p 3SAT \Rightarrow L \in NP \Rightarrow coNP \subset NP$.
Таким образом, $NP = coNP$.

Пусть $NP = coNP$. Тогда $TAUT \in coNP = NP \Rightarrow TAUT \leq_p 3SAT$ т.к. $3SAT \in NPC$. Аналогично $3SAT \in NP = coNP \Rightarrow 3SAT \leq_p TAUT$ т.к. $TAUT \in coNPC$.

№3

Не верно, т.к. в таком случае на словах из $\overline{L}$ машина Тьюринга может работать более чем полиномиальное время или вообще не остановиться т.е. $\overline{L}$ не разрешим за полиномиальное время, значит L тоже не разрешим за полиномиальное время.

№4

Рассмотрим алгоритм разрешения T:

```
def inT(n):
        m = 1
        while m < n:
                m = m * 3
        return m == n
```

Все операции (умножение на 3 и сравнения) выполняются за полиномиальное время от $len(m)$ (длины слова m), $len(m) \leq len(3n) \leq len(n) + 1$. Цикл выполнится ровно $k = \lceil \log_3(n) \rceil = \lceil \frac{\log_{10}(n)}{\log_{10}(3)} \rceil$ раз, $\log_{10}(n) \leq len(n)$. Таким образом, алгоритм работает за полиномиальное время от длины входа.

№5

Если в графе есть гамильтонов цикл, то последовательность вершин, образующих этот цикл, будет сертификатом: длина такого сертификата полиномиальна от длины входного слова (длина цикла равна числу вершин) и можно за полиномиальное время проверить, что такой цикл действительно есть в графе просто пройдя по нему. И наоборот: если существует сертификат – такая последовательность не повторяющихся вершин, что каждая из них смежна со следующей и последняя смежна с первой, то в графе очевидно есть гамильтонов цикл.

№6

Если таблица как-то заполнена, то можно за полиномиальное время проверить, что она заполнена нужным образом: для каждой из n^4 клеток проверить, что такого числа нет в строке (n^2 клеток), столбце (n^2 клеток) и малом квадрате (n^2 клеток). Правильное заполнение таблицы будет сертификатом: оно полиномиально от размера входа и оно же будет решением судоку. Таким образом, это задача класса NP, значит $SUDOKU \leq_p SAT$ т.к. $SAT \in NPC$.