Филиппова Анастасия

Домашнее задание 10

1

$N = 391, p = 17, q = 23, e = 3.$
$1)d \cdot e = 1 \ mod \ 352,$
$3d + 353y = 1,$
$3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ 352 \begin{bmatrix} 0 \\ 1 \end{bmatrix},$
$1 \begin{bmatrix} -117 \\ 1 \end{bmatrix} \ 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix},$
$d = -117 + 352k$, при $k = 1$ получаем $d = 235$.
$2)m^e = 41^3 = 117 \cdot 41 = 105 \ mod \ 391.$
$3)(m^e)^d = 105^{235} \ mod \ 391,$
$235 = 128 + 64 + 32 + 8 + 2 + 1,$
$105^1 = 105 \, mod \, 391, 105^2 = 77 \, mod \, 391, 105^8 = 186 \, mod \, 391, 105^{32} = 154 \, mod \, 391, 105^{64} = 256 \, mod \, 391, 105^{128} =$
$239 \ mod \ 391,$
$105 \cdot 77 \cdot 186 \cdot 154 \cdot 256 \cdot 239 = 41 \ mod \ 391.$

2

3

Найдем $d:$
$2021 = 43 \cdot 47 \Longleftrightarrow \varphi(2021) = 1932,$
$25d + 1932y = 1,$
$25 \begin{bmatrix} 1 \\ 0 \end{bmatrix} 1932 \begin{bmatrix} 1 \\ 0 \end{bmatrix},$
$7 \begin{bmatrix} -77 \\ 1 \end{bmatrix} 25 \begin{bmatrix} 1 \\ 0 \end{bmatrix},$
$4 \begin{bmatrix} 3 \cdot 77 + 1 \\ -3 \end{bmatrix} 7 \begin{bmatrix} -77 \\ 1 \end{bmatrix},$
$3 \begin{bmatrix} -77 \\ 1 \end{bmatrix} 4 \begin{bmatrix} 1 + 3 \cdot 77 \\ -3 \end{bmatrix},$
$1 \begin{bmatrix} 7 \cdot 77 + 2 \\ -7 \end{bmatrix},$
Получаем: $d = 541_1 932k \Longleftrightarrow d = 541$. Ответ: 541.

4

(i) Будем делать что-то на подобии бинарного поиска. На каждой итерации алгоритма будем смотреть, является ли центральный элемент горкой. Если да, то нашли требуемое. Если нет, то сравниваем центральный элемент с соседними и если он меньше левого, то вызываем алгоритм на левом подмассиве, если меньше правого, вызываемся от правого подмассива. Таким образом находим горку.
Докажем корректность. Пусть мы вызвались от подмассива, в котором не было горки. Это значит, что все элементы подмассива больше того элемента, с которым мы сравнивали центральный на предыдущей итерации. То есть массив монотонный. То есть его конец будет являться горкой. Следовательно, получаем противоречие. То есть, если в массиве есть горка, то мы точно ее нйдем.
Алгоритм использует $O(\log n)$ сравнений, так как на каждой итерации производим 2 сравнения. Количество итераций равно количество итреций бинарного поиска на данном массиве.
(ii) Рассмотрим массив, содержащий одну горку. То есть: $a[0] < a[1] < ... < a[k-1] < a[k] > a[k+1] > ... > a[n-1]$. Докажем, что $\log n$ — нижняя оценка на число сравнений. Дейсвительно, пусть мы сравниваем элемент массива $a[i]$ с элементом массива $a[j]$, $(i < j)$ тогда в случае, если $a[i] < a[j]$, то горка

находится либо между $a[i]$ и $a[j]$., либо справа от жлемента $a[j]$. В худшем случае длина массива $i+1...n$ — массив, в котором находится горка, будет больше чем $1....i$ массив, в котором горку не ищем. Таким образом, при каждом сравнение мы в худшем случае откидываем точно не больше половины массива. Отсюда получаем нижнюю оценку на число сравнений: $\Omega(\log n)$. То есть для любого n существует массив, на котором получим ответ, сделав такое количество сравнений.

5

Если граф G принадлежит описанному в условии языку, то его дополнение является двудольным графом. Действительно: дополнение к полному подграфу есть множество несмежных между собой вершин. Алгоритм: выбираем вершину, запускаем обход в глубину. Совешаем обход в глубину и красим вершины в 2 цвета следующим образом: допустим, мы пошли в первую вершину — помечаем её как 0. Затем просматриваем все смежные вершины, и если не помечена вершина, то на ней ставим пометку 1 и рекурсивно переходим в нее. Если же она помечена и на ней стоит та же пометка, что и у той, из которой шли (в нашем случае 1), значит граф не является двудольным. Следовательно, исходный граф не принадлежит языку.
Корректность: доказательство корректности на прямую следует из того, что граф является двудольным тогда и только тогда, когда он 2−раскрашиваемый. А это вроде бы очевидно.
Следовательно, определить, является ли дополнение графа двудольным графом можем за полином.
Следовательно, данный язык принадлежит классу P, а так как P не равно NP, то данный язык $\notin NPc$.

6

Рассмотрим описанный в условии язык. Ребра и вершины в пути могут повторяться. Из этого можем сделать вывод, что если между вершинами s и t существует хотя бы какой-то путь, то есть t достижима из s, то тройка (G, s, t) принадлежит языку. Существование пути из s в t проверяем, запустив из вершины s BFS или DFS.
Сложность:
$O(|E|\log|V|) = O(|V|^2\log|V|) = O(|V|^4) \longleftrightarrow$ алгоритм полиномиален по длине входа.
Следовательно, описанный в условии язык, принадлежит к классу P. Так как $P \in coNP \iff$ язык принадлежит $coNP$.

7

Отсортируем массив, в котором хранятся ребра графа с их весами за $O(m\log m)$. Далее будем действовать аналогично алгоритму бинарного поиска: 1) рассматриваем ребро (u, v),
2) удаляем это ребро и все ребра, которые находятся слева от этого ребра в отсортированном массиве (ребра, вес которых меньше чем вес рассматриваемого ребра). Далее с помощью BFS проверяем, стал ли граф двудольным. Если получили двудольный граф, то рассматриваем массив слева от рассматриваемого ребра, если нет — справа.
Докажем корректность алгоритма: пусть сложность текущей раскраски равна a, то есть наибольший вес ребра, соединяющего 2 вершины одного цвета, равен a. Тогда удалим это ребро и все ребра легче данного. Тогда мы получим двудольный граф. То есть при бин поиске мы каждый раз будем смотреть, можно ли сделать трудность раскраски меньше, удаляя соотвтсвующие веса и проверяя, является ли граф двудольным.
Сложность: сортировка массива за $O(m\log m)$, BFS за $O(|V| + |E|) = O(m)$.
$T(n) = T(n/2) + O(m)$, получаем сложность $O(m\log m)$.