Домашнее задание №6

Гунаев Руслан, 776 группа

9 апреля 2019 г.

Подготовка к midterm.

1.

Пусть $k = 2n + 1$– нечетное. Будем строить дерево рекурсии.

0) 1.

1) n.

2) n^2.

3) $n^2(n-1)$

4) $n^2(n-1)^2$

$2k+1-3)$ $\dfrac{(n!)^2}{2}$

$2k)$ $(n!)^2$

Видим, что на последних 3 уровнях имеем асимптотику $(n!)^2$.

$$\sum_{i=0}^{n-2} 2((n)!/(n-i-1)!)^2/((n-i)) - \frac{(n!)^2}{2} + 4(n!)^2 \leqslant f(n) \leqslant \sum_{i=0}^{n-2} 2((n)!/(n-i-1)!)^2 + 3(n!)^2$$

Воспользовавшись тем, что функции возрастающие, посчитали суммы, взяв интеграл от функций, стоящих под знаком суммирования. Получилось так

$$\sum_{i=0}^{n-2} 2((n)!/(n-i-1)!)^2/((n-i)) - \frac{(n!)^2}{2} + 4(n!)^2 = 10\pi n \left(\frac{n}{e}\right)^{2n}$$

$$\sum_{i=0}^{n-2} 2((n)!/(n-i-1)!)^2 + 3(n!)^2 = 10\pi n \left(\frac{n}{e}\right)^{2n}$$

Замечание: вышло так, что данные интегралы равнялись в точности последнему члену суммы.

Аналогично сделаем для четного k.

$$T(k) = 10\pi\lfloor k/2\rfloor \left(\frac{\lfloor k/2\rfloor}{e}\right)^{2\lfloor k/2\rfloor}$$

2.

1)

Пусть $f(n) = \log n, g(n) = n$. Понятно, что $f(n) = O(n), g(n) = \Omega(n)$.

$$f(n) \cdot g(n) = n\log n, \forall C > 0 \exists N : \forall n \geqslant N \to n\log n < Cn^2 \Rightarrow f(n) \cdot g(n) \neq \Omega(n^2)$$

Нет.

2)

Пусть $f(n) = \cos\frac{\pi n}{2} + 1, g(n) = \sin\frac{\pi n}{2} + 1$, понятно, что $f(n) \geqslant, g(n) \geqslant 0$. Очевидно, что для любой константы найдется $n_1 : f(n_1) < Cf(n_1), n_2 : f(n_2) > Cf(n_2)$, причем этих чисел будет бесконечно много в силу периодичности функций.

3.

1)

$$\log n! \sim n \log n, \text{ по формуле Стирлинга.}$$

$$\log n = O(n^c), \forall c > 0 \Rightarrow n \log n = O(n^{\log_{2017} 2018 - \varepsilon}), \varepsilon = \frac{1 - \log_{2017} 2018}{2}$$

По основной теореме о рекуррентах получим, что

$$T(n) = \Theta(n^{\log_{2017} 2018})$$

2)

$$T(n) = n^2 \sum_{k=1}^{\log n - 1} \frac{1}{\log n - k}$$

Взяв интеграл, получим

$$T(n) = \Theta(n^2 \log\log n)$$

4.

1)

$$T(n) = T(n-1) + n\sqrt{n}$$

$$T(n) = \sum_{i=0}^{n} (n-i)^{3/2}, T(n) \leqslant n^{5/2}$$

$$T(n) \geqslant \frac{n}{2}\left(\frac{n}{2}\right)^{3/2} = \Omega(n^{5/2}) \Rightarrow T(n) = \Theta(n^{5/2}).$$

2)

$$T(n) = \sqrt{n} T(\sqrt{n}) + n$$

Возьмем производную этой функции.

$$T'(x) = \frac{T(\sqrt{x})}{2\sqrt{x}} + \frac{T'(\sqrt{x})}{2} + 1$$

Значение функции в нуле равно 0. Значение в точке $\varepsilon, \varepsilon \to 0$ точно больше нуля, значит больше нуля производная фукнции в точке 0. Значит больше нуля производная функции в любой точке(по индукции).

В силу доказанного выше факта ограничим n степенями двойки.

$$2^{k-1} \leqslant n \leqslant 2^k$$

Построив дерево рекурсии, легко понять, что на каждом уровне дерева мы тратим $O(n)$ времени. Значит всего потратим

$$T(2^k) = \log k 2^k$$

Так как $n = 2^k, \sqrt{n} = 2^{k/2}$, то уровней в дереве будет $\log k$.

Аналогично проделаем с меньшей степенью двойки, в итоге получим, что

$$T(n) = \Theta(n \log\log n).$$

5.

$$E(f-g)^2 = \sum p_i(f^2(x_i) - 2f(x_i)g(x_i) + g^2(x_i))$$

Положим, $g(x) = kx + b$. Раскроем скобки, получим кучу знаков суммирования и индексов, которые мне очень лень техать, а главное получим функцию от двух переменных. Найдем ее точки экстремума.

$$h'_k = \sum p_i y_i x_i + k \sum p_i x_i^2 + b \sum p_i x_i = 0$$

$$h'_b = \sum p_i y_i + k \sum p_i x_i + b = 0$$

Выразим $b(k)$, подставим в первое уравнение, получим единственную точку, она и будет точкой минимума.

$$k = \frac{\sum p_i y_i x_i - \sum p_i y_i \sum p_i x_i}{\left(\sum p_i x_i\right)^2 - \sum p_i x_i^2}$$

Подставим это в выражение для $b(k)$. Это итоговый ответ.

6.

Возьмем $L = \Sigma^*$, тогда если $N = \Sigma^* \setminus R, R$– неразрешимый язык, то N не лежит в $co - RP$. Каждый язык, лежащий в NP является разрешимым, значит любой неразрешимый язык не лежит в NP. Так как $L \in P$, то $L \in co - NP, N \subset L, N \notin co - NP$.

7.

Язык $EUCLID \in P$, язык $NOHAMPATH \in co - NPC$, так как $HAMPATH \in NPC$. Если $NOHAMPATH \leqslant_m EUCLID$, то $\forall L \in co - NP \to L \in P \Rightarrow co - NP \subset P \Rightarrow co - NP = P$.

Пусть $L \in NP \Rightarrow \overline{L} \in co - NP, P \Rightarrow \overline{L} \in NP$. Так как класс P замкнут относительно дополнения, то $L \in P \Rightarrow NP \subset P \Rightarrow P = NP = co - NP$.

8.

Построим сводимость $CLIQUE \leqslant_m GAR$. Пусть на вход ам подается граф из n вершин, нам нужно определить, если ли в нем клика размера k. Будем идти по вершинам графа и строить новый граф так, чтобы каждой вершине соответсвовала клика размера n. Если между двумя вершинами есть ребро, то проведем ребро из соответствующих этим вершинам клик в новом графе. Из одной вершины старого графа выходит не более $n-1$ ребра, поэтому соединяя клики между собой, будем выбирать в них вершины, из который еще не выходят ребра. Покажем, что

сводимость действительно полиномиальна. В новом графе будет n^2 вершин и E ребер, сколько и было в старом графе. Построение нового графа займет не более $O(En + n^2)$ операций.

Покажем, что сводимость корректна. Если в старом графе была клика размера k, то заменив, в новом графе каждый подграф на вершину, получим ту же клику размера k. Пусть в старом графе нет клики, значит построенная конфигурация нам не даёт эквивалентной клики, будем искать другую и составлять новые подграфы из n вершин каждая, но этого мы сделать не можем по построению Из каждой вершины в старом графе выходит не более n рёбер, поэтому новый граф будем строить так: если между двумя вершина есть ребро, то в соответствующих подграфах выбираем "свободные» вершины и соединяем их. Новая конфигурация должна отличаться от старой хотя бы на одну вершину, но тогда эта вершина может быть соединена максимум с одной из вершин в новой конфигурации, а значит эта конфигурация не будет давать полный граф.

В качестве сертификата можем привести k множеств из n вершин каждое. Длина сертификата равна $O(V), V-$ количество вершин в графе. Сначала нам надо будет проверить, что вершины в каждом из множеств попарно связны, что делается за $O(kn^2)$. Далее нам надо будет проверить, что для любых двух множеств существует соединяющее их ребро, это мы можем сделать(грубая оценка) за $O(k^2n^2)$. Будем идти по вершинам из множеств, смотреть их ребра и искать вершину, из которой также выходит это ребро, в других множествах.

Таким образом, $GAR \in NPC$.

9.

$$f(n) = \omega(n^c), f(n) = o(2^{nd})$$

Прологарифмируем выражения, получим

$$\log f(n) > c \log c_1 n, \log f(n) < ndc_2$$

Положим $\log f(n) = \log^2 n$, проверим.

$$\frac{n^c}{n^{\log n}} = \frac{1}{n^{\log n - c}} \to 0 \forall c > 0.$$

$$\frac{2^{\log^2 n}}{2^{nd}} = 2^{\log^2 n - nd} \to 0 \forall d > 0$$

Таким образом, $f(n) = 2^{\log^2 n}$.

10.

Пусть дано слово $\omega : |\omega| = n$. Построим полный граф на $n + 1$ вершине. Уберем ребро (i, j), если $\omega[i; j)$ не принадлежит языку. Для этого есть сертификат непринадлежности, так как язык лежит в классе $co - NP$. В итоге получим граф. Если не существует путь из первой вершину в $n-$ую, то и слово не принадлежит итерации языка. В качестве сертификата непринадлежности подаем сертификаты для каждого (i, j),а также сам граф. Очевидно, этот сертификат полиномиален от длины входа, в силу того, что каждый сертификат непринадлежности полиномиален.

11.

Пусть $L \in NPC \cap co - NP$.

$$\forall L' \in NP \to L' \leqslant_m L, L \in NPC \Rightarrow \overline{L} \in co-NPC \Rightarrow L \leqslant_m \overline{L} \Rightarrow L' \leqslant_m \overline{L} \Rightarrow L' \in co-NP$$

$$\forall L' \in co-NP \to L' \leqslant \overline{L}, L \in co-NP \Rightarrow \overline{L} \in NP \Rightarrow \overline{L} \leqslant_m L \Rightarrow L' \leqslant_m L \Rightarrow L' \in NP$$

Получили, что $NP \subset co-NP, co-NP \subset NP \Rightarrow NP = co-NP$.

12.

$$d_1 = 3, d_2 = 8.$$

Раскладывая определитель по столбцу получим

$$d_n = 3d_{n-1} - d_{n-2}$$

Решим данную рекурренту.

$$\lambda_{1,2} = \frac{3 \pm \sqrt{5}}{2}$$

Использовав начальные условия и положив $d_0 = 0$, получим

$$d_n = \frac{1}{\sqrt{5}} \left(\left(\frac{3+\sqrt{5}}{2} \right)^{n+1} + \left(\frac{3-\sqrt{5}}{2} \right)^{n+1} \right) \Rightarrow d_n = \Theta \left(\left(\frac{3+\sqrt{5}}{2} \right)^{n} \right)$$

$4^2 \equiv 5(mod 11)$, поэтому заменим $\sqrt{5}$ на 4. Также учитывая, что $4 \cdot 3 \equiv 1, 2 \cdot 6 equiv 1$, получаем

$$d_{2018} = 3(42)^{2019} - 3(60)^2 019 = 3(-2)^{2019} - 3(5)^{2019}$$

$$a^{10} = 1(mod 11)$$

Из-за этого получим

$$d_{2018} = 3(-2)^9 - 3(5)^9 = 10$$

Основные задания.

13.

Заведем матрицу и вектор соответсвтущий каждому элементу последовательности.

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_{k-1} & a_k \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \mathbf{F_n} = \begin{pmatrix} F_n \\ F_{n-1} \\ \vdots \\ F_{n-k+1} \end{pmatrix}$$

Заметим, что

$$\mathbf{F_{n+1}} = A\mathbf{F_n} \Rightarrow \mathbf{F_{n+t}} = A^t \mathbf{F_n}$$

$$\mathbf{F_k} = \begin{pmatrix} F_k \\ F_{k-1} \\ \vdots \\ F_1 \end{pmatrix}$$

А значит,

$$\mathbf{F_n} = A^{n-k}\mathbf{F_k}$$

14.

$$d_n = C_1\left(\frac{a-\sqrt{a^2+4b}}{2}\right)^n + C_2\left(\frac{a+\sqrt{a^2+4b}}{2}\right)^n$$

$$d'_n = C_1\left(\frac{a-x}{2}\right)^n + C_2\left(\frac{a+x}{2}\right)^n, x^2 = a^2+4b(modp)$$

Рассмотрим $d_n - d'_n$. Покажем, что у этой последовательности коэффициенты равны 0.
Рассмотрим

$$(a-b)^n + (a+b)^n,$$

заметим, что в таком случае коэффициенты при нечетных степенях b сократятся, а при четных увеличатся вдвое. Если $b = \sqrt{a^2+4b}$, то $b^{2k} \equiv ((\sqrt{a^2+4b})^2)^k \equiv x^{2k}(modp)$
Заметим тогда, что $d_n - d'_n \equiv 0(modp)$

15.

$$(u_1+u_2x)(v_1+v_2x) = (u_1v_1+u_2v_2a) + (u_1v_2+v_1u_2)x$$

16.

Аналогично 14 задаче раскроем $(a-x)^n + (a+x)^n$.
Рассмотрим $x^{2k+1} = xx^{2k} = x(a^2+4b)^k$. $x^{2k} = (a^2+4b)^k$. Значит

$$(a-x)^n + (a+x)^n = A_n + B_nx$$

Но мы знаем, что все коэффициенты при нечетных степенях x обнуляются, а значит, $B_n = 0$. Что и требовалось доказать.

17.

Пусть $a = td, b = kd, gcd(k,t) = 1 \Rightarrow b-a = d(k-t)$. Покажем, что $gcd(k, k-t) = 1$, пусть равно $c \neq 1$. Тогда $k \vdots c \Rightarrow k-t \vdots c \Rightarrow t \vdots c \Rightarrow gcd(k,t) = c \neq 1$. Противоречие, получили, что

$$gcd(a,b) = gcd(a, b-a).$$

18.

Нахождение остатка числа при делении на другое, также сложение двух чисел делается за полином от длин входа. Нам нужно найти количество рекурсивных вызовов в алгоритме Евклида. По теореме Ламе следует, что если $a < F_{k+1}$, то алгоритм сделает не более k рекурсивных вызовов. Пусть $F_k \leqslant a < F_{k+1} \Rightarrow F_{k+1} < 2a \Rightarrow k < \log 2a$. Итоговая асимптотика $O(\log b \log a)$

19.

Поделим уравнение на 2. Воспользуемся расширенным алгоритмом Евклида.

a	b	d	x	y
114	161	1	-24	17
47	114	2	17	-7
20	47	2	-7	3
7	20	2	3	-1
6	7	1	-1	1
1	6	6	1	0
0	1		0	1

Так как мы решили для $ax + by = gcd(a, b)$, то умножим решения на 5.

Получим

$$x = -120, y = 85.$$

20.

$$ax + by = 0$$

Поделим a, b на их общий делитель. Получим

$$a'x + b'y = 0$$

Если x является решением, то оно обязано делиться на b', так как для любых $yb'y$ делится на b'. Аналогично y должен делиться на a'.

Тогда решение исходного уравнения будет иметь вид

$$x = \frac{b}{gcd(a, b)}, y = \frac{-a}{gcd(a, b)}.$$

Для прошлой задачи общее решение будет следующим

$$x = 161t - 120, y = -114t + 85.$$

21.

$$n \equiv r(mod m_1 m_2) \Rightarrow n = r + t m_1 m_2 \Rightarrow n \equiv r = r_1(mod m_1), n \equiv r = r_2(mod m_2).$$

22.

$$\begin{cases} x \equiv 7(mod 24) \\ x \equiv 3(mod 28) \\ x \equiv 10(mod 21) \end{cases} \Leftrightarrow \begin{cases} x = 7(8) \\ x = 1(3) \\ x = 3(7) \end{cases}$$

По китайской теореме об остатках получим.

$$x = 31.$$

23.

$$10^{25^{10}} \bmod 92.$$

$92 = 4 \cdot 23$. Заметим, что $10^{25^{10}}$ сравнимо с 0 по модулю 4. $\varphi(23) = 22, gcd(10, 23) = 1$. $25 = 3(mod22) \Rightarrow 25^{10} = 3^{10} = 1(mod22)$.

$$10^{25^{10}} = 10^{22t+1} = 10(mod23)$$

Применим китайскую теорему об остатках

$$x = 0 + 10 \cdot 4 \cdot 6 = 240 = 56(mod92).$$