Домашнее задание № 1

Гунаев Руслан, 776 группа

21 февраля 2019 г.

Основные задачи.

1.

1) Пусть $f(n) \in O(g(n))$. Тогда $\exists C > 0 : \exists N > 0 : \forall n \geqslant N \hookrightarrow f(n) \leqslant Cg(n)$.

2) Пусть $f(n) \notin \omega(g(n)) \Rightarrow \exists C > 0 : \forall N \exists n > N \hookrightarrow f(n) \leqslant Cg(n)$.

3) Пусть $g(n) \notin o(f(n)) \Rightarrow \exists C > 0 : \forall N \exists n > N \hookrightarrow Cf(n) \leqslant g(n)$.

4) Пусть $g(n) \in \Omega(f(n)) \Rightarrow \exists C > 0 : \exists N : \forall n > N \hookrightarrow Cf(n) \leqslant g(n)$

Видим, что 1 и 4 условия эквивалентны, так как для константы из первого условия можем взять $1/C$ из 4, аналогично в обратную сторону. То же самое работает и для 2 и 3.

Если выполняется условие 1), то выполнется и условие 2), ведь для выполнения 2) мы можем взять ту же константу, так же $\forall N \exists n = N + 1$, это верно в силу выполнения 1) условия.

Аналгочно, если выполняется 4),то выполняется 3).

Покажем, что 1 и 2 условия эквивалентны в случае неотрицательности и монотонности функций(неубывания).

Покажем, что если выполняется отрицание 1 условия, то выполняется и отрацание 2. Пусть $f(n) \notin O(g(n)) \Rightarrow \forall C > 0 \exists N \forall n > N \hookrightarrow f(n) > Cg(n)$. Но тогда не может выполняться условие $f(n) \notin \omega(g(n)) \Rightarrow \exists c > 0 : \forall N \exists n > N \hookrightarrow f(n) \leqslant cg(n)$. А значит $f(n) \in \omega(g(n))$. Получили, что 1 и 2 условия эквивалентны. А значит $4 \iff 1 \iff 2 \iff 3$(функии неотрицательные и неубывающие).

Рассмотрим случай, что когда функции могут быть отрицательными и немонотонными.

$$f(n) = \cos n, g(n) = \sin n$$

Заметим, что 2) и 3) условия выполняются, но понятно, что не выполняются 1) и 4).

Если убрать ограничения с функций, то получим

$$1 \Rightarrow 2 \iff 3 \Leftarrow 4, 1 \iff 4$$

2.

1.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \leqslant \frac{n^k}{k!}$$

При достаточно болших n будем считать, что $n \geqslant 2k \Rightarrow (n-k) \geqslant n/2$. Получим следующую оценку.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \geqslant \frac{n^k}{2^k k!}$$

В результате

$$\binom{n}{k} \in \Theta(n^k)$$

2.

$$\binom{n^8+2n^4}{n^4} = \frac{(n^8+2n^4)!}{n^4!(n^8+n^4)!} \propto \frac{\sqrt{n^8+2n^4}(n^8+2n^4)^{n^8+2n^4}}{\sqrt{(n^8+n^4)n^4}n^{4n^4}(n^8+n^4)^{n^8+n^4}} \sim$$
$$\sim \frac{n^4 n^{8n^8+16n^4}}{n^6 n^{8n^8+12n^4}} = \Theta(n^{4n^4-2})$$

3.

Рассмотрим три случая.

а) $n = 3k$.

$$\binom{6k}{k} \sim \frac{\sqrt{2\pi 6k}(6k)^{6k}e^{6k}}{e^{6k}\sqrt{5}2\pi k(5k)^{5k}} \Rightarrow \binom{6k}{k} = \Theta\left(\left(\frac{6^6}{5^5}\right)^{n/3}\frac{1}{\sqrt{n}}\right)$$

б, в)

Два оставшихся случая друг от друга существенно не отличаются. $n = 3k+a, a = \{1, 2\}$.

$$\sqrt{(}2\pi(6k+a)) = \Theta(\sqrt{k})$$
$$(6k+a)^{6k+a} = (6k+2)^{6k}(6k+2)^2 = \Theta(k^{6k}k^2 6^{6k})$$

k^2 присутствует как в числителе, так и в знаметеле, поэтому получим:

$$\binom{2n}{\lfloor \frac{n}{3} \rfloor} = \Theta\left(\left(\frac{6^6}{5^5}\right)^{n/3}\frac{1}{\sqrt{n}}\right)$$

3.

1)

Использовав свойство, доказанное на семинаре, получим:

$$\int_1^n \frac{1}{x}dx \leqslant \sum_{k=1}^{n}\frac{1}{k} \leqslant \int_1^{n+1}\frac{1}{x}dx$$

$$\ln k\Big|_1^n \leqslant \sum_{k=1}^{n}\frac{1}{k} \leqslant \ln(k+1)\Big|_1^{n+1}$$

$$\ln n \leqslant \sum_{k=1}^{n}\frac{1}{k} \leqslant \ln(n+1) \Rightarrow \sum_{k=1}^{n}\frac{1}{k} = \Theta(\ln n)$$

2)

Ограничим сумму сверху:

$$\sum_{k=1}^{n}\ln k \leqslant n\ln n = O(n\ln n)$$

Ограничим сумму снизу:

$$\sum_{k=1}^{n}\ln k = 0 + \ln 2 + \ln 3 + \ln 4 + ... + \ln\frac{n}{2} + ... + \ln n \geqslant$$
$$\geqslant \left\lfloor\frac{n}{2}\right\rfloor \cdot \ln\left(\left\lfloor\frac{n+1}{2}\right\rfloor\right) \geqslant \left(\frac{n}{2}-1\right)\cdot\ln\left(\frac{n+1}{2}-1\right) =$$
$$= \left(\frac{n}{2}-1\right)\ln(\frac{n}{2}-\frac{1}{2}) \geqslant \frac{n}{100}\cdot\ln n = \Omega(n\ln n)$$

Получим:

$$\sum_{k=1}^{n}\ln k = \Theta(n\ln n)$$

4.

1) Воспользуемся тем, что функция неубывающая.

$$\int_1^n 2^{\sqrt{x}}dx \leqslant \sum_{k=1}^{n}2^{\sqrt{k}} \leqslant \int_1^{n+1}2^{\sqrt{x}}dx$$

$$\int 2^{\sqrt{x}}dx = \frac{(2\,\log 2\,\sqrt{x}-2)\;e^{\log 2\,\sqrt{x}}}{(\log 2)^2}$$

$$\sum_{k=1}^{n}2^{\sqrt{k}} = O(\sqrt{n+1}e^{\sqrt{n+1}})$$

$$\sum_{k=1}^{n}2^{\sqrt{k}} = \Omega(\sqrt{n}e^{\sqrt{n}}))$$

Таким образом получим, что

$$\sum_{k=1}^{n} 2^{\sqrt{k}} = \Theta(\sqrt{n}e^{\sqrt{n}})$$

2) Очевидно, что

$$\sum_{k=1}^{n} 2^{k^2} \leqslant \sum_{k=1}^{n^2} 2^{k}$$

так как таким образом мы пробегаемся по всем квадратам от 1 до n^2. С другой стороны

$$\sum_{k=1}^{n} 2^{k^2} \geqslant 2^{n^2}$$

Воспользуемся тем, что 2^n неубывающая функция.

$$\int_1^{n^2} 2^x dx \leqslant \sum_{k=1}^{n^2} 2^k \leqslant \int_1^{n^2+1} 2^x dx \Rightarrow \sum_{k=1}^{n^2} 2^k \sim 2^{n^2}$$

В результате получим:

$$\sum_{k=1}^{n} 2^{k^2} \in \Theta(2^{n^2})$$

5.

Рассмотрим функцию.

$$f(n) = \tan(\frac{\pi}{2}\cos 1/n)$$

Нетрудно показать, что данная функция $f(x)$ является строго возрастающей при $x \geqslant 1$.

Рассмотрим рекурренту:

$$T(n) = 2T(n/2) + f(n)$$

Заметим, что

$$\lim \frac{n^{\log_2 2}}{f(n)} = 0 \Rightarrow f(n) = \Omega(n)$$

Проверим условие регулярности, которое должно выполняться для любых достаточно больших n и некоторого $k < 1$.

$$2\tan(\frac{\pi}{2}\cos 2/n) \geqslant k\tan(\frac{\pi}{2}\cos 1/n)$$

Очевидно, это не является правдой, ведь

$$\lim \frac{\tan(\frac{\pi}{2}\cos 1/n)}{\tan(\frac{\pi}{2}\cos 2/n)} = 1 \Rightarrow \exists N : \forall n \leqslant N \hookrightarrow S = \frac{\tan(\frac{\pi}{2}\cos 1/n)}{\tan(\frac{\pi}{2}\cos 2/n)} \leqslant 3/2$$

Но

$$2 = kS \leqslant 3/2.$$

Противоречие, значит условие регулярности не выполняется.

6.

В силу того, что функция возрастающая, можем ограничить n близжайшими степенями двойки. $2^{m-1} \leqslant n \leqslant 2^m$. Заметим, что формулировка задачи эквивалентна следующей записи:

$$T(n) = n(T(n/2) + O(1))$$

Начнем раскрывать скобки:

$$\begin{gathered}
T(n) = n(n/2(n/4...(T(1) + O(1)) + O(1)) + ... + O(1)) \leqslant \\
\leqslant 2^m(2^{m-1}...(T(1) + O(1)) + O(1)) + ... + O(1)) = [T(1) = O(1)] = \\
= O(2^{m+m-1+...+1}) + O(2^{m+m-1+...+2}) + ... + O(2^m) = \\
= [2^{m+m-1+...+1} = M] = \\
= O(M) + O(M/2) + O(M/8) + ... + O(1) = O(M + M/2 + M/8 + ... + 1) \leqslant \\
\leqslant O(M + M/2 + M/4 + ... + 1) = \\
= O(M) = O(2^{m(m+1)/2}) = O\left(n^{\frac{\log_2 n+1}{2}}\right)
\end{gathered}$$

С другой стороны вся выписанная выше сумма больше любого своего слагаемого, так как все слагаемые положительные, поэтому найдем максимальное слагаемое.

$$a_{max} = n^{\frac{\log_2 n+1}{2}}$$

Аналогично проделаем для $n = 2^{m-1}$.

Таким образом, получим:

$$T(n) = \Theta\left(n^{\frac{\log_2 n+1}{2}}\right)$$

7.

В силу того, что функция возрастающая, можем ограничить n близжайшими степенями двойки. $2^{m-1} \leqslant n \leqslant 2^m$. Перепишем условие:

$$T(n) = 2T(n/2) + \frac{n}{\log n}$$

Будем строить дерево рекурсии.

0-ой уровень: $\dfrac{n}{\log n}$ операций.

1-ый уровень: $2 \cdot \frac{n}{2} \cdot \frac{1}{\log n/2} = \dfrac{n}{\log n/2}$

Далее аналогично. Получим сумму:

$$T(n) = n \sum_{i=0}^{\log_2 n-1} \frac{1}{\log_2 \frac{n}{2^i}}$$

Заменим сумму на интеграл, как делали в прошлых задачах. Сначала посчитаем для $n = 2^m$.

$$T(n) \leqslant 2^m \int_0^{m-1} \frac{1}{k-x} dx = 2^m \log m$$

Аналогично посчитаем для $n = 2^{m-1}$.

В итоге получим, что

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

8.

Пусть $f(n) = n^3$.

$$f(n) = C_1 f(n-1) + C_2 f(n-2) + C_3 f(n-3) + C_4 f(n-4)$$

Раскроем скобки, получим:

$$\begin{gathered} C_1 + C_2 + C_3 + C_4 = 1 \\ 3C_1 + 6C_2 + 9C_3 + 12C_4 = 0 \\ 3C_1 + 12C_2 + 27C_3 + 48C_4 = 0 \\ C_1 + 8C_2 + 27C_3 + 64C_4 = 0 \end{gathered}$$

Решив уравнения, получаем $C_1 = 4, C_2 = -6, C_3 = 4, C_4 = -1$

Итого:

$$f(n) = 4f(n-1) - 6f(n-2) + 4f(n-3) - f(n-4)$$

9.

$$f_{n+1} = 3f_n + 4f_{n-1} + n^2$$

Найдем общее решение, решив характеристическое уравнение.

$$\lambda^2 - 3\lambda - 4 = 0 \Rightarrow \lambda_1 = 4, \lambda_2 = -1 \Rightarrow f_n^c = C_1 4^n + C_2(-1)^n$$

Частное решение будем искать в виде $An^2 + Bn + C$. Подставим в рекурренту, получим значения коэффициентов.

$$A = -1/6, B = -5/18, C = -4/27$$

Получим:

$$f_n = C_1 4^n + C_2(-1)^n - 1/6n^2 - 5/18n - 4/27$$

10.

$$f_{n+1} + f_n + f_{n-1} = 0$$

Решим характеристическое уравнение:

$$\lambda^2 + \lambda + 1 = 0 \Rightarrow \lambda_{1,2} = \frac{-1 \pm i\sqrt{3}}{2}$$

Получим общее решение:

$$f_n = C_1 e^{2i\pi n/3} + C_2 e^{-2i\pi n/3} = C_3 \cos\frac{2\pi n}{3} + C_4 \sin\frac{2\pi n}{3}$$

11.

Пусть есть числа $\{a_1, a_2, ..., a_k\}$ общей длиной n. Разобьем числа на пары и будем перемножать. Если два числа имеют разную длину, то дополним меньшее нулями справа, после умножения уберем эти нули, это займет $O(|a_i|)$ времени.

Посчитаем, сколько времени займет умножение чисел в каждой паре в сумме. Будем считать, что умножение происходит по методу Карацубы. Обозначим это время $S(n)$.

$$S(n) \leqslant O(|a_1|^{\log_2 3}) + ... + O(|a_k|^{\log_2 3})) \leqslant O(\sum_{i=1}^{k} |a_i|^{\log_2 3}) \leqslant O(n^{\log_2 3})$$

После такого произведения получится $k/2$ чисел. Заметим, что:

$$|a_i \cdot a_j| \leqslant |a_i| + |a_j|$$

С новыми числами проделаем то же самое. Учтем, что общая длина всех новых чисел не превосходит n. Итого получим, что данный алгоритм $T(n)$ будет занимать следующее время:

$$T(n) = O(\log(k) n^{\log_2 3})$$

12.

Построим дерево рекурсии.

0-ой уровень: $\Theta(n)$.

1-ый уровень: $\Theta(n/5) + \Theta(7n/10) = \Theta(9n/10)$.

2-ой уровень: $\Theta(81n/100)$.

Продолжим аналогично и просуммируем.

$$T(n) = n + n \sum_{i=1}^{p} \left(\frac{9}{10}\right)^i$$

p – высота дерева. Заметим, что

$$T(n) \geqslant n, T(n) \leqslant 11n \Rightarrow T(n) = \Theta(n)$$

Посчитаем для случаев, когда разбиваем на группы по 3 и по 4. Поиск медианы медиан на каждом этапе алгоритма будет занимать $T(n/4)$. Найдем количество элементов не превосходящих медиану медиан.

$$N_1 \geqslant 1/2(2n/4) = n/4$$

Получим

$$T(n) = T(n/4) + T(3n/4) + \Theta(n)$$

Максимальная высота дерева будет равна $\log_{4/3} n$, минимальная – $\log_3 n$. Таким образом, так как на каждом уровне дерева алгоритм будет занимать $\Theta(n)$ операций, то

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

Аналогично посчитаем для 3. Посчитаем число элементов не превосходящих медиану медиан.

$$N_2 \geqslant 1/2(2n/3) = n/3$$

Получим

$$T(n) = T(n/3) + T(2n/3) + \Theta(n)$$

Аналогично предыдущему случаю

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