АиСД у2024. Первый семестр

Домашние задания M3138-M3139

⟨Версия от 11 сентября 2024 г.⟩

Темы

Неделя 1. Время работы алгоритма. О-нотация

Устная часть

1.1. Докажите по определению, что если $f_1(n) = \mathcal{O}(g_1(n))$ и $f_2(n) = \mathcal{O}(g_2(n))$, то $f_1(n) + f_2(n) = \mathcal{O}(g_1(n) + g_2(n))$, если $f_i(n)$ и $g_i(n)$ неотрицательны.

1.2. Докажите по определению, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$, если $f(n)$ и $g(n)$ неотрицательны.

1.3. Докажите по определению, что $\sum\limits_{i=1}^{n+5} 2^i = \mathcal{O}(2^n)$.

1.4. Докажите по определению, что $\frac{n^3}{6} - 7n^2 = \Omega(n^3)$.

1.5. Докажите по определению, что $2^n \cdot n^2 = \mathcal{O}(2.1^n)$.

1.6. Докажите по определению, что $\log_3(n^2 + 5n + 3) = \Theta(\log n)$.

1.7. Докажите или опровергните следующие утверждения. Для опровержения достаточно привести пример, на котором утверждение неверно. Все функции положительные.

(a) Если $f(n) = \mathcal{O}(g(n))$, то $g(n) = \mathcal{O}(f(n))$

(b) Если $f(n) = \mathcal{O}(g(n))$, то $2^{f(n)} = \mathcal{O}(2^{g(n)})$

(c) $f(n) = \mathcal{O}(f(n)^2)$

(d) $f(n) = \Theta(f(\frac{n}{2}))$

(e) Если $f(n) = \mathcal{O}(g(n))$, то $g(n) = \Omega(f(n))$

1.8. Пусть $p(n) = \sum\limits_{i=0}^{d} a_i n^i$, где $a_d > 0$ — полином степени d от n. Используя определения, полагая, что k — константа, докажите, что:

(a) Если $k \geq d$, то $p(n) = \mathcal{O}(n^k)$

(b) Если $k = d$, то $p(n) = \Theta(n^k)$

(c) Если $k \leq d$, то $p(n) = \Omega(n^k)$

1.9. Придумайте две положительные функции $f(n)$ и $g(n)$, такие что не выполнено ни одно из двух: $f(n) = \mathcal{O}(g(n))$ и $f(n) = \Omega(g(n))$.

1.10. Докажите или опровергните, что $\log(n!) = \Theta(n \log n)$.

1.11. Докажите или опровергните, что $\sum\limits_{i=1}^{n} \frac{n}{i} = \mathcal{O}(n \log n)$.

1.12. Докажите, что функции $T_1(n) = a \cdot T_1\left(\frac{n}{b}\right) + n^c$ и $T_2(n) = a \cdot T_2\left(\frac{n}{b}\right) + d \cdot n^c$ имеют одинаковую Θ-оценку при любых константах a, b, c и d.

1.13. Докажите по индукции, что если $T(n) = 2T(\sqrt{n}) + 1$, то $T(n) = \mathcal{O}(\log n)$. В этом и следующих заданиях в качестве базы индукции можно полагать, что $T(1) = 1$.

1.14. Докажите по индукции, что если $T(n) = 3T\left(\frac{n}{2}\right) + 1$, то $T(n) = \Omega(n)$.

1.15. Докажите по индукции, что если $T(n) = \log_2 n \cdot T\left(\frac{n}{\log_2 n}\right) + n$, то $T(n) = \mathcal{O}(n \log n)$.

1.16. Докажите по индукции, что если $T(n) = 2T\left(\frac{n}{2} + \log_2 n\right) + n$, то $T(n) = \mathcal{O}(n \log n)$.

1.17. Докажите по индукции, что если $T(n) = 2T\left(\frac{n}{2} + 20\right) + n$, то $T(n) = \mathcal{O}(n \log n)$.

1.18. Найдите Θ-асимптотику времени работы алгоритма, построив дерево рекурсии, если $T(n) = T\left(\frac{n}{3}\right) + \log_2(n)$.

1.19. Найдите Θ-асимптотику времени работы алгоритма, построив дерево рекурсии, если $T(n) = T\left(\frac{n}{2} + \sqrt{n}\right) + \sqrt{6046}$.

1.20. Найдите Θ-асимптотику времени работы алгоритма, построив дерево рекурсии, если $T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + n$.

1.21. Пусть время работы алгоритма A равно $T_A(n) = 7T_A\left(\frac{n}{2}\right) + n^2$, а время работы алгоритма B равно $T_B(n) = aT_B\left(\frac{n}{4}\right) + n$. При каких значениях a второй алгоритм работает асимптотически быстрее первого?