16 października 2024

Zadania z matematyki dyskretnej, lista nr 3

1. Niech $f(n) = \sum_{k=1}^{n} \lceil \log_2 k \rceil$. Wykaż, że
$$f(n) = n - 1 + f(\lceil n/2 \rceil) + f(\lfloor n/2 \rfloor)$$
dla wszystkich $n \geq 1$. Pokaż, że jeśli w powyższej zależności wymagamy, by $f(1) = 0$, to f jest jedyną funkcją spelniającą tę zależność.

   Wsk.: Rozbij $\sum \lceil \log_2 k \rceil$ na sumy po k parzystych i nieparzystych

2. Wyznacz zwartą (tzn. bez symboli „$\sum$" i „$\cdots$") postać funkcji $f(n)$ z poprzedniego zadania.

3. Przedstawieniem liczby naturalnej n w układzie liczb Fibonacciego nazywamy taki ciąg współczynników $a_2, \ldots, a_k$ równych 0 lub 1, że $a_2F_2 + \cdots a_kF_k = n$ oraz $a_i + a_{i+1} \leq 1$ dla wszystkich i. Pokaż, że każda liczba naturalna ma jednoznaczne przedstawienie w układzie liczb Fibonacciego.

4. Niech x, k, n będą liczbami całkowitymi. Skonstuuj algorytm obliczający x^k modulo n. Algorytm powinien korzystać z wzorów: $x^{2l} = x^l \cdot x^l$, $x^{2l+1} = x \cdot x^{2l}$. Określ liczbę mnożeń wykonywanych przez ten algorytm.

5. Znajdź wzór na n-tą potęgę macierzy $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Napisz szybki algorytm wyliczający z tego wzoru n-tą liczbę Fibonacciego F_n. Oszacuj złożoność tego algorytmu, jeśli używa on procedury mnożącej dwie liczby k-cyfrowe w czasie $M(k)$ (załóż, że $M(k) \geq 2M(k/2)$).

6. Pokaż, że
$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} F_2 \\ F_1 \end{pmatrix} = \begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix}.$$
Skonstruuj algorytm który dla danych $p_1, \ldots, p_k, a_1, \ldots, a_k \in \mathbb{R}$ wylicza szybko a_n zadane zależnością rekurencyjną $a_n = p_1 a_{n-1} + p_2 a_{n-2} + \cdots + p_k a_{n-k}$.

7. Skonstruuj algorytm mnożący dwie n-cyfrowe liczby całkowite a i b poprzez rozbicie każdej z nich na trzy równe części, a_1, a_2, a_3 i b_1, b_2, b_3. Powinien się on zadowalać pięcioma wywołaniami rekurencyjnymi samego siebie obliczającymi: $m_1 = a_1 \cdot b_1$, $m_2 = a_3 \cdot b_3$, $m_3 = (a_1 + a_2 + a_3) \cdot (b_1 + b_2 + b_3)$, $m_4 = (a_1 - a_2 + a_3) \cdot (b_1 - b_2 + b_3)$, $m_5 = (a_1 - 2a_2 + 4a_3) \cdot (b_1 - 2b_2 + 4b_3)$. Oszacuj złożoność obliczeniową $T(n)$ tego algorytmu.

   Wsk.: Skorzystaj z tego, że liczbę długości n można podzielić przez niewielką stałą w czasie $O(n)$ (jak?).

8. Dany jest algorytm typu „dziel i zwyciężaj" wywołujący sam siebie (rekurencyjnie) a razy dla podproblemów rozmiaru n/b i wykonujący poza tym $c \cdot n^d$ operacji. Czas $T(n)$ działania takiego algorytmu spełnia zależność rekurencyjną: $T(n) = aT(n/b) + cn^d$. Korzystając z tej zależności oszacuj $T(n)$ jako $O(\cdot)$ w zależności od a, b, d. Możesz założyć, że $n = b^k$.

   Wsk.: Rozważ trzy przypadki: $a > b^d, a = b^d, a < b^d$

9. Niech
$$T(n) \leq T(\lceil n/5 \rceil + 2) + T(\lceil 7n/10 \rceil + 2) + cn$$
Pokaż używając indukcji, że $T(n) < c'n$ dla pewnej stałej c'.

10. Udowodnij, że jeśli $a, b \in \mathbb{N}$ oraz x i y są niezerowymi liczbami całkowitymi spełniającymi równanie diofantyczne $ax + by = 1$, to
$$\gcd(a, b) = \gcd(a, y) = \gcd(x, b) = \gcd(x, y).$$
Wykaż ponadto, że dokładnie jedna z liczb x i y musi być ujemna.

11. (a) Przedstaw $\gcd(448, 721)$ w postaci $721x + 448y$, dla $x, y \in \mathbb{Z}$.
    (b) Oblicz takie całkowite x, y, że $333x + 1234y = 1$. Ile się równa 333^{-1} w pierścieniu $\mathbb{Z}_{1234}$.
    (c) Oblicz -69^{-1} mod 1313.

12. Pokaż, że $\gcd(F_{n-1}, F_n) = 1$. Udowodnij indukcyjnie, że $\gcd(F_m, F_n) = F_{\gcd(m,n)}$.

13. Udowodnij, że jeśli $a \perp b$ i $a > b$, to $\gcd(a^m - b^m, a^n - b^n) = a^{\gcd(m,n)} - b^{\gcd(m,n)}, 0 \leq m < n$.

14. Pokaż, że dla każdego n istnieje dokładnie jedna potęga 2, która w układzie dziesiętnym ma n cyfr z najbardziej znaczącą cyfrą 1.

15. Niech $ax_0 + by_0 = c$ dla pewnych $a, b, c, x_0, y_0 \in \mathbb{Z}$. Określ zbiór wszystkich rozwiązań (x, y) równania
$$ax + by = c.$$