Барицький Андрій ПМО-21 21.06.22

Екзамен з теорії алгоритмів

Білет № 26

1. Обчислювальні функції. Найпростіші функції.

Обчислювальними функціями називають числові функції, значення яких можна підраховувати за допомогою деякого алгоритму, який для заданої функції є єдиним.

Нехай N — множина всіх натуральних чисел і $N^{(n)} = \{<x_1, \ldots, x_n> \mid x_i \in N\}$ — множина всіх можливих n натуральних чисел.

Тоді: функцією наступності ($S^1(x)$) називають числову функцію $\varphi: N \to N$, якщо $\varphi(x) = x + 1$. Тотожно рівною ($O^n(x_1, \ldots, x_n)$) нулю називають числову функцію $\varphi: N^{(n)} \to N$, якщо $\varphi(x_1, \ldots, x_n) = 0$. Функцією вибору аргумента ($I_i^n(x_1, \ldots, x_n)$) називають числову функцію $\varphi_i: N^{(n)} \to N$, якщо вона повторює значення свого i-го аргумента: $\varphi_i(x_1, \ldots, x_n) = x_i$ ($1 \le i \le n$). Такі функції (O^n, I_i^n, S^1) називають найпростішими.

2. Важкорозв'язні задачі. Поліноміальні алгоритми та важкорозв'язні задачі.

Якщо алгоритм потребує не більше $T(n)$ операцій на вході довжини n, то йому буде необхідно не більше $P(T(n))$ операцій для таких самих вхідних даних в двійковому записі, де P — поліноміальна функція, яка визначає зростання кількості операцій при переході до двійкових операцій.

Отже, поліноміальність чи експоненціальність часу роботи алгоритму інваріантна, тобто вона незмінкована, бо $P(T(n))$ обмежене зверху деяким поліномом відносно n.

Більшість задач мають поліноміальні алгоритми розв'язування. Такі задачі називають легкорозв'язними. Важкорозв'язними називають задачі, для яких не існує поліноміальних алгоритмів для їх розв'язування.

Для експоненціальних алгоритмів важливо знати, наскільки „погано" вони працюють, щоб мати уявлення про те, чи можливо в принципі дочекатися на завершення його виконання.

В той же час поліноміальні задачі вважають „практично розв'язними", бо на практиці вони працюють досить швидко, обсяг цього класу практично не залежить від вибору конкретної моделі обчислень і клас таких задач має природні властивості замкнутості. До того ж композиція двох поліноміальних алгоритмів також працює за поліноміальний час.

3. Довести, що функція $f(x_1, x_2) = x_1 \cdot x_2$ є примітивно рекурсивною.

$$f(x_1, x_2) = x_1 \cdot x_2$$

$$g(x_1) = f(x, 0) = x \cdot 0 = 0 = O^1(x_1)$$

$$h(x_1, x_2, f(x_1, x_2)) = (x_1, x_2 + 1) = x_1(x_2 + 1) = x_1 \cdot x_2 + x_1 =$$

$$= S^3(+, f(x_1, x_2), x_1) = S^3(+, f(x_1, x_2), I^3_1(x_1, x_2, f(x_1, x_2)));$$

$$h(x_1, x_2, y) = S^3(R(I^1_1(x_1), S^1(I^3_3(x_1, x_2, y))), I^3_3(x_1, x_2, y), I^3_1(x_1, x_2, y)).$$