Теми за държавен изпит

6. Сложност на алгоритъм. Асимптотично поведение на целочислени функции (O-, Ω-, Θ-, o-, ω-нотация). Сложност на рекурсивни програми.

def] Детерминирана машина на Тюринг, или МТ, е наредена седморка $\langle \Sigma, \Gamma, Q, q_0, q_y, q_n, \delta \rangle$, където:

- Σ е крайна азбука (непразна), наречена входна азбука
- $\Gamma \supset \Sigma$ е крайна непразна азбука, наречена лентова азбука, като символът "$\sqcup$" за празна клетка е в $\Gamma \setminus \Sigma$.
- Q е крайно множество от състояния
- $q_0 \in Q$ е началното състояние
- $q_y \in Q$, $q_y \neq q_0$, е приемащото състояние
- $q_n \in Q$, $q_n \neq q_0$, е отхвърлящото състояние
- $\delta : (Q \setminus \{q_y, q_n\}) \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow, \downarrow\}$ е функцията на преходите

Машината разполага с безкрайна лента в двете посоки, разбита на клетки, а във всяка клетка се съдържа точно една буква от Γ.

def] Нека M е МТ с вход x. Във всеки момент от работата на M, конфигурацията на M е наредената тройка (q, y, i), където $q \in Q$ е текущото състояние, $y \in \Sigma^*$ е съдържанието на лентата и $i \in \mathbb{Z}$ е текущата позиция на главата.
Приемаща конфигурация е всяка конф., в която $q = q_y$, а отхвърляща конфигурация е всяка конф., в която $q = q_n$.

Понятието алгоритъм е първично. За него няма общоприета прецизна формална дефиниция. Въпреки това са правени опити да се формализира понятието алгоритъм чрез най-различни модели на изчисления, като например Машина на Тюринг, машина с произволен достъп, език за програмиране и други. Според тезиса на Чърч всички те са еквивалентни, т.е. всеки нов изчислителен модел се оказва еквивалентен на известните такива (напр. МТ).

def| Изчислителна задача Π е двуместна релация $\Pi \subseteq \mathcal{I} \times \mathcal{S}$, където $\mathcal{I}$ е безкрайно множество от екземпляри, а $\mathcal{S}$ е крайно или безкрайно множество от решения. За всеки екземпляр $x \in \mathcal{I}$ и за всяко решение $y \in \mathcal{S}$, такова че $(x, y) \in \Pi$, казваме, че y е решение за x.

def| Нека е дадена изчислителна задача $\Pi \subseteq \mathcal{I} \times \mathcal{S}$. Нека $*$ е елемент, непринадлежащ на $\mathcal{S}$. Нека $\mathcal{S}' = \mathcal{S} \cup \{*\}$. Тогава всяка функция $f: \mathcal{I} \to \mathcal{S}'$ наричаме функция, съответна на Π, ако $\forall x \in \mathcal{I}$:

- ако съществува решение за x, то $f(x)$ е някое решение
- ако не съществува решение за x, то $f(x) = *$.

При дадена изчислителна задача Π, която е релация, изобразяваща екземпляри в решения, алгоритъм за Π е всяка реализация на функция, съответна на Π. Тази реализация се състои от крайна редица от елементарни инструкции.

def За произволен алгоритъм, големина на входа е:

- Ако алгоритъмът е върху числен масив, тогава е броят на числата в масива
- Ако алгоритъмът е върху графи, тогава е сумата от броя на върховете и ребрата (валидно както за графи без тегла, така и за тегловни)
- Ако алгоритъмът е върху низове, тогава е броят на символите.

def Нека Π е изчислителна задача и A е алгоритъм за нея. За всяка големина на входа $n \in \mathbb{N}^+$, нека $\mathcal{I}(n)$ е крайното множество от съществено различните входове с големина n. За всеки вход k, нека f(k) е броят стъпки, които се изпълняват от A(k). Тогава, за всяко n, сложността по време на A в най-лошия случай е

$$T_A(n) = \max\{f(k) \mid k \in \mathcal{I}(n)\},$$

а средната сложност по време на A е

$$Q_A(n) = \frac{1}{|\mathcal{I}(n)|} \sum_{k \in \mathcal{I}(n)} f(k).$$

Сложността по памет на A върху даден вход е броят на елементите памет, които A ползва, без да броим паметта, в която се разполага входът, и паметта, в която се разполага изходът. Сложността по памет в най-лошия и средния случай се дефинира аналогично на тази по време.

def Нека $f: \mathbb{R}^+ \to \mathbb{R}$. Казваме, че f е асимптотично положителна, ако $\exists n_0 \in \mathbb{R}^+ \, \forall n \geq n_0 : f(n) > 0$.

def! За всяка функция $g(n)$:

$$\Theta(g(n)) := \{ f(n) \mid \exists c_1, c_2 > 0 \; \exists n_0 > 0 \; \forall n \geq n_0 : \; 0 \leq c_1 . g(n) \leq f(n) \leq c_2 . g(n) \}.$$

$$O(g(n)) := \{ f(n) \mid \exists c > 0 \; \exists n_0 > 0 \; \forall n \geq n_0 : \; 0 \leq f(n) \leq c . g(n) \}$$

$$\Omega(g(n)) := \{ f(n) \mid \exists c > 0 \; \exists n_0 > 0 \; \forall n \geq n_0 : \; 0 \leq c . g(n) \leq f(n) \}$$

$$o(g(n)) := \{ f(n) \mid \forall c > 0 \; \exists n_0 > 0 \; \forall n \geq n_0 : \; 0 \leq f(n) < c . g(n) \}$$

$$\omega(g(n)) := \{ f(n) \mid \forall c > 0 \; \exists n_0 > 0 \; \forall n \geq n_0 : \; 0 \leq c . g(n) < f(n) \}$$

Свойства:

$f(n) = \Theta(f(n))$

$f(n) = \Theta(g(n)) \rightarrow g(n) = \Theta(f(n))$

$f(n) = \Theta(g(n)) \wedge g(n) = \Theta(h(n)) \rightarrow f(n) = \Theta(h(n))$

Нека дефинираме релация $\asymp$ по следния начин:

$f(n) \asymp g(n)$ т.с.т.к. $f(n) = \Theta(g(n))$.

Спрямо горните свойства, $\asymp$ е релация на еквивалентност.

За всяка функция $g(n)$, $O(g(n)) \cap \Omega(g(n)) = \Theta(g(n))$.

За всяка функция $g(n)$:

- $o(g(n)) \cap \omega(g(n)) = \emptyset$
- $O(g(n)) \cap \omega(g(n)) = \emptyset$
- $o(g(n)) \cap \Omega(g(n)) = \emptyset$

За всички функции $f(n)$, $g(n)$:

$f(n) = O(g(n)) \leftrightarrow g(n) = \Omega(f(n))$

$f(n) = o(g(n)) \leftrightarrow g(n) = \omega(f(n))$

За всеки две функции $f(n)$ и $g(n)$:

- $f(n) = o(g(n)) \rightarrow f(n) = O(g(n))$
- $f(n) = \omega(g(n)) \rightarrow f(n) = \Omega(g(n))$

За всеки две функции $f(n)$ и $g(n)$ и $\forall k \in \mathbb{R}^+$:

$$f(n) = \Theta(g(n)) \leftrightarrow (f(n))^k = \Theta((g(n))^k)$$

Нека $f(n) = g(n) + h(n)$, където f е асимптотично положителна, а за h имаме $|h(n)| = o(g(n))$. Тогава $f(n) = \Theta(g(n))$.

Нека $p(n)$ е произволен асимптотично положителен полином от степен k. Тогава $p(n) = \Theta(n^k)$.

Нека $f(n)$ и $g(n)$ са растящи и неограничени функции и $a > 1$ е константа. Тогава $f(n) = o(g(n)) \rightarrow a^{f(n)} = o(a^{g(n)})$. Също така, $\lg f(n) = o(\lg g(n)) \rightarrow f(n) = o(g(n))$.

Нека $f(n)$ и $g(n)$ са растящи и неограничени функции и нека $a > 1$ е константа. Тогава $a^{f(n)} = \Theta(a^{g(n)}) \rightarrow f(n) = \Theta(g(n))$. Също така, $f(n) = \Theta(g(n)) \rightarrow \lg f(n) = \Theta(\lg g(n))$.

Гранични теореми:

За всички функции $f(n)$, $g(n)$:

- $f(n) = o(g(n)) \leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$

- $f(n) = \omega(g(n)) \Longleftrightarrow \lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

- ~~[illegible]~~ $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = c, \xrightarrow{!} f(n) = \Theta(g(n))$, където $0<c<\infty$

<u>Мастър теорема:</u>

Нека $a \geq 1$ и $b > 1$ са константи и нека $f(n)$ е положителна функция. Нека $T(n) = a.T(\frac{n}{b}) + f(n)$, където $\frac{n}{b}$ има смисъл или на $\lfloor \frac{n}{b} \rfloor$, или на $\lceil \frac{n}{b} \rceil$. Тогава асимптотиката на $T(n)$ е следната:

<u>Случай 1:</u> Ако $f(n) = O(n^{\log_b a} / n^{\varepsilon})$ за някоя положителна константа ε, то $T(n) = \Theta(n^{\log_b a})$.

<u>Случай 2:</u> Ако $f(n) = \Theta(n^{\log_b a})$, тогава $T(n) = \Theta(n^{\log_b a} . \lg n)$.

<u>Случай 3:</u> Ако са изпълнени следните условия:

1. $f(n) = \Omega(n^{\log_b a} . n^{\varepsilon})$ за някоя положителна константа ε
2. съществува константа c, такава че $0 < c < 1$ и

$\exists n_0 \forall n \geq n_0 : a . f(\frac{n}{b}) \leq c . f(n)$,

то $T(n) = \Theta(f(n))$.

Условие 3.2 се нарича условие за регулярност.