Първо контролно по ДАА

Красимир Манев

Задачи I

Зад. I.1. Когато се изисква да се установи в какво отношение са порядъците на растеж на две дадени функции, не е необходимо да се правят проверки за петте дефиниции, тъй като установяването на едно от отношенията определя някои от другите. В случая е есествено да предположим, че $n! \in o((n+1)!)$. За целта трябва или да пресметнем $\lim_{n\to\infty} \frac{n!}{(n+1)!} = 0$, което е елементарно и доказва хипотезата или да приложим дефиицията. Ще покажем, че $\forall c > 0, \exists n_0$ такова, че $\forall n \geq n_0, 0 \leq n! \leq c(n+1)!$. Лявото неравенство е очевидно. Търсим n_0 такова, че $n_0! \leq c(n_0+1)!$. Значи $n_0 \geq \frac{1}{c} - 1$. След като сме доказали $n! \in o((n+1)!)$, автоматично поучаваме $n! \in O((n+1)!)$, $(n+1)! \in \omega(n!)$, $(n+1)! \in \Omega(n!)$ и $(n+1)! \notin \Theta(n!)$.

Зад. I.2. В тази задача са допуснати най-много грешки. Най-неприятната е свързана с погрешното разсъждение, че щом $h(n) = \min(f(n), g(n))$, то или $h(n) = f(n)$, или $h(n) = g(n)$. Разсъждението е валидно за скаларни стойности, но не и за функции и затова решения свързани с него са неприемливи. Най-простото решение е, да се отбележи, че от $h(n) \leq f(n), \forall n$ и $h(n) \leq g(n), \forall n \Rightarrow 2.h(n) \leq f(n) + g(n)$, т.е. при $c = 1/2, n_0 = 0, h(n) \leq c(f(n) + g(n)), \forall n \geq n_0$. Практически всички, които са се опитали да използват $c = 1$ са го направили по лош начин. Някои направо пишат невeрното $h(n) \leq f(n) + g(n)$ (например $f(0) = -1, g(0) = -1, h(0) = -1, f(0)+g(0) = -2$). Други се позовават на асимптотическата положителност на функциите но го правят "повърхностно". Редно е да се разпише явно какво означава асимптотически положителна т.е $\exists n_f, f(n) > 0, \forall n \geq n_f$ и $\exists n_g, g(n) > 0, \forall n \geq n_g$. Сега $h(n) \leq f(n) + g(n), \forall n \geq \max(n_f, n_g)$, което доказа твърдението с $c = 1$.

Зад. I.3. В тази задача неприятно впечатление правят опитите да се извършват оценки "на око", както и да се пресмятa производната на $\log n!$, което често води до погрешна оценка за асимптотичното поведение на $\frac{\log n!}{\log n}$. Доброто решение тук е да се замени $n!$ с приближението на Стирлинг, при което $\frac{\log n!}{\log n} \approx n$. При пресмятане на границата $\lim_{n\to\infty} \frac{\log n!}{\log^3 n}$ е допуснато следното грешно прилагане на теоремата за границата на сума, което може да бъде много опасно:

$$\lim_{n\to\infty} \frac{\log n!}{\log^3 n} = \lim_{n\to\infty} \frac{\log 1}{\log^3 n} + \lim_{n\to\infty} \frac{\log 2}{\log^3 n} + \ldots + \lim_{n\to\infty} \frac{\log n}{\log^3 n} = 0.$$

Не може да се прилага теоремата за граница на сума, когато сумирането е направено по променливата n, която клони към безрайност.

Задачи II

Зад. II.1. Типична грешка в решението на тази задача е неправилното определяне на корените на РО - от хомогенната част получаваме характеристично уравнение $x^2 - 2 = 0$, с корени $x_1 = \sqrt{2}$ и $x_2 = -\sqrt{2}$. От нехомогенната част $2^{n/2} = (\sqrt{2})^n.1$ получаваме още един корен $x_3 = \sqrt{2}$. Двойният корен $\sqrt{2}$ дава решението $O(n.2^{n/2})$.

Зад II.2. Мастър-теоремата, доказана на лекции, не е приложима, когато нехомогенната част $d(n)$ не е мултипликативна функция. Може да има версия на мастър-теоремата, при която това не е важно, но дори да е така или тази версия не дава добра оценка или е приложена неправилно, за да се получи често срещаната оценка $O(n^{\log_2^4}) = O(n^2)$. Всъщност мастър-теоремата е приложима за немултипликативна функция $d(n) = c.h(n)$, където c е положителна константа, а $h(n)$ е мултипликативна. В такъв случай обаче се сравняват a и $h(b)$ и тъй като в случая имаме $h(n) = n, a = 3, b = 2$ е изпълнено $a > h(b)$ и решението е $O(n^{\log_2^3})$.

Зад II.3. Много студенти не са успели да осъществят тривиалната итерация $T(2) = 2^2, T(3) = 2^4, T(4) = 2^8, ..., T(n) = 2^{2^{n-1}}$. Малко от направилите правилната хипотеза са я доказали с индукция (което, формално погледнато, е задължително). Някои студенти са заместили полученото $O(2^{2^{n-1}})$ с $O(2^{2^n})$, което формално не е грешно, но е безсмислено, защото $2^{2^{n-1}} \in o(2^{2^n})$

Задачи III

Зад III.1. След като първият цикъл се изпълнява за $i = 1, 2, \ldots, n^2$, а втория за $j = 1, 2, \ldots, i$, то тялото на втория цикъл (което има константна сложност) ще се изпълни $1 + 2 + \ldots + n^2 = O(n^4)$ пъти.

Зад III.2. След като първият цикъл се изпълнява за $i = 1, 2, \ldots, n/2$, а втория за $j = 1, 1 + i, 1 + 2i, \ldots, 1 + \lfloor n/i \rfloor i$, то тялото на втория цикъл (което има констанна сложност) ще се изпълни $\frac{n}{1} + \frac{n}{2} + \ldots + \frac{n}{n/2} = n(\frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{n/2}) = O(n. \log n)$ пъти, защото $\frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{n/2}$ е сума на първите няколко члена на хармоничния ред, клоняща към $\ln n$ при n клонящо към безкрайност.

Зад III.3. Рекурсивната функция директно ни води до рекурентно отношение от вида $T(n) = 4T(n-3) + O(1)$ с характеристично уравнение $x^3 - 4 = 0$. Заблуда е, че това уравнение има троен корен $x = 4^{1/3}$ и решението е $O(n^2.4^{n/3})$. Коренът е единичен (другите два корена са комплексни числа) и затова решението е $O(4^{n/3})$.

Задача IV

Бързото решение (при такива m, че може да се разположи масив с m елемента) е: “линейно сортиране” и преглеждане на сортирания масив

- сложност $O(n+m)$. Ако m е много голямо а i малко, алтернативата е да се използва пирамида, в която да се държат най-малките i от прегледаните до момента елементи, която за всеки нов по-малък е обновява за $O(\log i)$ - сложност $O(n. \log i)$. Сортиране, дори със сложност $O(n. \log n)$ е неприемливо.