Решения на контролното по ДАА от 23.04.2016 г.

Зад. 1 Нека $f(n)$, $g(n)$, $\phi(n)$ и $\psi(n)$ са произволни положителни функции, дефинирани върху $\mathbb{N}^+$. Докажете или опровергайте всяко от тези твърдения:

3 т а) $f(n) = \Theta(g(n)) \wedge \phi(n) = \Theta(\psi(n)) \quad \rightarrow \quad f(n) \cdot \phi(n) = \Theta(g(n) \cdot \psi(n))$

7 т б) $f(n) = \Theta(g(n)) \wedge \phi(n) = \Theta(\psi(n)) \quad \rightarrow \quad f(n)^{\phi(n)} = \Theta\left(g(n)^{\psi(n)}\right)$

Решение: Твърдение а) е вярно:

$$
\begin{aligned}
&f(n) = \Theta(g(n)) \wedge \phi(n) = \Theta(\psi(n)) \leftrightarrow \\
&\begin{cases} \exists c_1, c_2, n_0\, \forall n \geq n_0 : c_1 g(n) \leq f(n) \leq c_2 g(n) \\ \exists c_3, c_3, n_1\, \forall n \geq n_1 : c_3 \psi(n) \leq \phi(n) \leq c_4 \psi(n) \end{cases} \rightarrow \\
&\exists c', c'', n_2\, \forall n \geq n_2 : c' g(n)\psi(n) \leq f(n)\phi(n) \leq c'' g(n)\psi(n)
\end{aligned}
$$

тъй като можем да вземем $c' = c_1 c_3$, $c'' = c_2 c_4$ и $n_2 = \max\{n_0, n_1\}$.

Твърдение б) не е вярно. Контрапример са, да кажем, $f(n) = 1$, $g(n) = 2$, $\phi(n) = \psi(n) = n$. $\square$

Зад. 2 За всяка от следните пет функции f_i, $1 \leq i \leq 5$, намерете съответна проста функция $g_i(n)$, такава че $f(n) \asymp g(n)$. Функцията да е проста означава да има къс, прост запис. Дайте кратка обосновка, като можете да ползвате наготово резултати, изведени на лекция.

$f_1(n) = \sum_{k=1}^{n} \frac{1}{2^k}$, $\quad f_2(n) = \sum_{k=1}^{n} \frac{1}{k}$, $\quad f_3(n) = \sum_{k=1}^{n} \left\lceil \frac{1}{k} \right\rceil$, $\quad f_4(n) = \sum_{k=1}^{n} \frac{1}{\lg k}$, $\quad f_5(n) = \lg n!$

Решение: $f(n) \asymp 1$, понеже $\sum_{k=1}^{n} \frac{1}{2^k} = 1$. $f_2(n)$ е n-тата парциална сума на хармоничния ред; на лекции бе доказано, че тя расте асимптотично като $\lg n$. Тъй като $\forall k_{\,1 \leq k \leq n} \left\lceil \frac{1}{k} \right\rceil = 1$, то $f_3(n) = \sum_{k=1}^{n} 1 = n$, така че $f_3(n) \asymp n$. $f_5(n) \asymp n \lg n$, факт, който е доказан на лекции чрез логаритмуване на апроксимацията на Стирлинг.

Оценяването на асимптотиката на $f_4(n)$ е значително по-трудно. Първо, тази дефиниция на $f_4(n)$ не е коректна, защото има събираемо с нула в знаменателя. Нека предефинираме $f_4(n)$ така:

$$f_4(n) \stackrel{\text{def}}{=} \sum_{k=2}^{n} \frac{1}{\lg k}$$

Следното решение е на асистент **Стефан Фотев**. Ще получим асимптотиката на сумата $\sum_{k=2}^{n} \frac{1}{\lg k}$, оценявайки интеграла $\int_2^n \frac{dx}{\ln x}$. Нека $F(n) = \int_2^n \frac{dx}{\ln x}$ и $G(n) = \int_2^n \frac{dx}{\ln^2 x}$. Ще интегрираме по части:

$$
\begin{aligned}
F(n) &= \int_2^n \frac{dx}{\ln x} = \left.\frac{x}{\ln x}\right|_2^n - \int_2^n x\, d\left(\frac{1}{\ln x}\right) = \frac{n}{\ln n} - \frac{2}{\ln 2} - \left(\int_2^n x(-1)\frac{1}{\ln^2 x}\frac{1}{x}\, dx\right) \asymp \frac{n}{\ln n} + \int_2^n \frac{dx}{\ln^2 x} \\
&= \frac{n}{\ln n} + G(n)
\end{aligned}
$$

Веднага следва $F(n) \succeq \frac{n}{\ln n}$.

Но за всички достатъчно големи n, $G(n) \leq \frac{1}{2}F(n)$, откъдето следва

$$F(n) \preceq \frac{n}{\ln n} + \frac{1}{2}F(n)$$

което влече $F(n) \preceq \frac{n}{\ln n}$.

И така, $F(n) \asymp \frac{n}{\ln n}$. $\square$

Зад. 3 Подредете по асимптотично нарастване следните шест функции, зададени чрез рекурентни уравнения. Обосновете колкото е възможно по-добре отговорите си. Напишете окончателната наредба в явен вид.

$$f_1(n) = 4T\left(\frac{n}{5}\right) + \lg n, \qquad f_2(n) = f_2\left(\frac{n}{5}\right) + f_2\left(\frac{4n}{5}\right) + 1, \qquad f_3(n) = f_3\left(\frac{n}{3}\right) + 3,$$

$$f_4(n) = f_4(n-1) + f_4(n-2), \qquad f_5(n) = f_5(n-1) + \frac{1}{n}, \qquad f_6(n) = f_6(\sqrt{n})$$

Решение: По първия случай на Мастър Теоремата, $f_1(n) \asymp n^{\log_5 4}$. С дърво на рекурсията, $f_2(n) \asymp n$ – това се извежда по начин, много близък до извеждането на лекция, че $T(n) = T(n/3) + T(2n/3) + n$ има решение $T(n) \asymp n \lg n$. По втория случай на Мастър Теоремата, $f_3(n) \asymp \lg n$. С метода са характеристичното уравнение, $f_4(n) \asymp \phi^n$, където $\phi = \frac{1+\sqrt{5}}{2}$. $f_5(n) = H_n$, където H_n е n-тата парциална сума на хармоничния ред; на лекции показахме, че $H_n \asymp \lg n$, така че $f_5(n) \asymp \lg n$. И накрая, при тази дефиниция, $f_6(n) = \Theta(1)$ по очевидни причини; ако $f_6(n) \asymp \lg\lg n + 1$, то $f_6(n) = \Theta(\lg\lg n)$, което показахме на лекция.

След като получихме асимптотиките на всички функции, с пет сравнения показваме, че наредбата е:

$$f_6 \prec f_5 \asymp f_3 \prec f_1 \prec f_2 \prec f_4$$

□

Зад. 4 Докажете чрез инварианта на цикъла, че Алгоритъм1 връща точно $\lfloor \log_2 n \rfloor + 1$.

```
Алгоритъм1(n ∈ ℕ⁺)
1  i ← 0
2  s ← 1
3  while s ≤ n do
4      s ← 2 × s
5      i ← i + 1
6  return i
```

Решение: Следното твърдение е инварианта за **while**-цикъла:

При всяко достигане на ред 3, $s = 2^i$.

База При първото достигане на ред 3 е изпълнено $i = 0$ заради присвояването на ред 1 и $s = 1$ заради присвояването на ред 2. Но тогава е изпълнено $s = 2^i$, защото $1 = 2^0$. ✓

Запазване Да допуснем, че за някое достигане на ред 3, което не е последно, е изпълнено $s = 2i$. Тогава, след присвояванията на редове 4 и 5, спрямо новите стойности на s и i отново е изпълнено $s = 2^i$.

Терминация Да разгледаме последното достигане на ред 3. Тогава $s > n$. Тъй като $s = 2^i$, вярно е, че $2^i > n \leftrightarrow i > \log_2 n$. Тогава очевидно $i > \lfloor \log_2 n \rfloor$. Понеже i е цяло число, то

$$i \in \{\lfloor \log_2 n \rfloor + 1, \lfloor \log_2 n \rfloor + 2, \lfloor \log_2 n \rfloor + 3, \ldots\}$$

Ще покажем, че i е равно на най-малкото от тези числа. Забелязваме, че $\frac{s}{2} \leq n$, защото при предишното достигане на ред 3, булевото условие е било изпълнено, така че старата стойност на s, а именно $\frac{s}{2}$ в термините на текущото s, не е надхвърляла n. Но

$$\frac{s}{2} \leq n \leftrightarrow \frac{2^i}{2} \leq n \leftrightarrow 2^{i-1} \leq n \leftrightarrow i - 1 \leq \log_2 n \leftrightarrow i \leq \log_2 n + 1$$

Веднага следва, че

$$i \notin \{\lfloor \log_2 n \rfloor + 2, \lfloor \log_2 n \rfloor + 3, \ldots\}$$

така че $i = \lfloor \log_2 n \rfloor + 1$.

□

Зад. 5 *Мода* на масив от числа се нарича всеки най-често срещан елемент на масива. Докажете, че не съществува алгоритъм, базиран на директни сравнения, който изчислява мода във време $o(n \lg n)$.

Решение: Ако можехме да изчисляваме мода чрез хипотетичен алгоритъм ALGX, базиран на директни сравнения, във време $o(n \lg n)$, то следният алгоритъм, базиран на директни сравнения, щеше да решава ELEMENT UNIQUENESS във време $o(n \lg n)$:

```
AlgY(A[1,...,n])
  k ← AlgX(A[])
  count ← 0
  for i ← 1 to n
      if A[i] = k
          count ← count + 1
  if count > 1
      return No
  else
      return Yes
```

Но на лекции сме доказали, че всеки алгоритъм за задачата ELEMENT UNIQUENESS, базиран на директни сравнения, има сложност $\Omega(n \lg n)$. Следователно, ALGX не съществува. □

Зад. 6 Разгледайте задачата, при дадени два сортирани масива $A[1,\ldots,n]$ и $B[1,\ldots,n]$, да се извърши *merge* на тези масиви; тоест, решението е масив с големина $2n$, състоящ се от елементите на $A[]$ и $B[]$, но в сортиран вид.

10 т а) Използвайки дърво на вземане на решения, докажете, че всеки алгоритъм за тази задача, който е базиран на директни сравнения, извършва поне $2n - o(n)$ сравнения на елемент от $A[]$ с елемент от $B[]$.

Решение Всички възможни изходи от *merge*-а са $\binom{2n}{n}$. Използвайки факта, че $\binom{2n}{n} = \frac{(2n)!}{n! \cdot n!}$ и използвайки три пъти апроксимацията на Стирлинг за трите факториела, лесно доказваме, че $\binom{2n}{n} \asymp \frac{1}{\sqrt{n}} 2^{2n}$. Всяко двоично дърво за вземане на решения би трябвало да има поне $\binom{2n}{n}$ листа, откъдето височината му е поне $\log_2 \binom{2n}{n}$. Използвайки това, че $\binom{2n}{n} \asymp \frac{1}{\sqrt{n}} 2^{2n}$, логаритмуваме израза вдясно при основа две и получаваме, че височината на дървото е не по-малка от $2n - o(n)$.

□

10 т б) Докажете, че всеки алгоритъм за тази задача, базиран на директни сравнения, по време на работата си сравнява всяка двойка елементи от $A[]$ и $B[]$, които са съседи в изхода.

Решение Да допуснем противното: съществува алгоритъм ALGX за тази задача, такъв че за поне едно n има вход $(A[1,\ldots,n], B[1,\ldots,n])$, такъв че $\text{ALGX}(A[], B[])$ не сравнява поне два елемента $A[i]$ и $B[j]$, които са съседи в *merge*-натия масив. Без ограничение на общността, нека $A[i] < B[j]$. Модифицираме $B[]$ така, че $B[j]$ бива намален с някакво положително ϵ, такова че сега отново $A[i]$ и $B[j]$ са съседи в *merge*-натия масив, но $B[j] < A[i]$. Нека така модифицираният масив вече се нарича $B'[]$. Пускаме $\text{ALGX}(A[], B'[])$. Но изходът от работата на $\text{ALGX}(A[], B'[])$ е същият като от работата на $\text{ALGX}(A[], B[])$, защото и в двата случая се вършат едни и същи сравнения (помним, че $A[i]$ не бива сравняван с $B[j]$, така че алгоритъмът не "усеща" модифицирането на $B[j]$, защото е базиран на директни сравнения – резултатите от всички други извършени сравнения са същите!) и в дървото на вземане на решения се минава по един и същи път от корена до някое листо. Веднага се вижда, че $\text{ALGX}(A[\,], B'[])$ не работи коректно, понеже слага $B[j]$ след $A[i]$. Следователно, ALGX не е коректен.

□

10 т в) Използвайки резултата от б), докажете долна граница $2n - 1$ за броя на сравненията на елемент от $A[]$ с елемент от $B[]$.

Решение Да разгледаме вход, при който изходният масив е

$$\big[A[1], B[1], A[2], B[2], \ldots, A[n], B[n]\big]$$

Очевидно такъв вход съществува за всяко n, така че този пример се мащабира за всяко n—ако докажем долна граница за този пример, тя е долна граница и за задачата.

Съгласно доказаното в подусловие б), алгоритъмът трябва да извърши поне $2n - 1$ сравнения, защото толкова са двойките съседни елементи в *merge*-натия масив, единият елемент на които е от $A[\,]$, а другият, от $B[\,]$.

$\square$