Решения на задачите от поправката на контролна работа № 1 по ДАА

I. 1) Определете в кои от петте отношения O, Ω, Θ, o и ω са функциите $f(n) = n5^n$ и $g(n) = \left(3^{n+\sqrt{n}}\right)((\lg n)!)$. Попълнете таблицата с „Да" или „Не", като *обосновете* отговорите си.

Решение:
Ако вземем логаритъм от двете функции, получаваме

$$\lg(f(n)) = \lg n + n \lg 5$$
$$\lg(g(n)) = n \lg 3 + \sqrt{n} \lg 3 + \lg((\lg n)!)$$

Тъй като $\lg((\lg n)!) = \Theta((\lg n)(\lg \lg n))$, ясно е, че асимптотиката на $\lg(g(n))$ се определя от линейния член. Следователно $\lg(f(n)) = \Theta(n)$ и $\lg(g(n)) = \Theta(n)$. От това не можем да направим заключение за нарастванията на $f(n)$ и $g(n)$. Иначе казано, опитът да решим задачата чрез директно логаритмуване се провали.

Ето едно възможно решение. Да разделим двете функции на 3^n. Получаваме

$$\frac{f(n)}{3^n} = n\left(\frac{5}{3}\right)^n$$
$$\frac{g(n)}{3^n} = \left(3^{\sqrt{n}}\right)((\lg n)!)$$

Нека $\frac{f(n)}{3^n} = f_1(n)$ и $\frac{g(n)}{3^n} = g_1(n)$. Да вземем логаритъм от двете.

$$\lg f_1(n) = \lg n + n \lg\left(\frac{5}{3}\right)$$
$$\lg g_1(n) = \sqrt{n} \lg 3 + \Theta((\lg n)(\lg \lg n))$$

Очевидно $\lg f_1(n) = \Theta(n)$ и $\lg g_1(n) = \Theta(\sqrt{n})$, следователно $\lg f_1(n) = \omega(\lg g_1(n))$, следователно $f_1(n) = \omega(g_1(n))$, следователно $f(n) = \omega(g(n))$. От това следва, че $f(n) = \Omega(g(n))$, $f(n) \neq O(g(n))$, $f(n) \neq o(g(n))$ и $f(n) \neq \Theta(g(n))$.

$f(n) = O(g(n))$	$f(n) = \Omega(g(n))$	$f(n) = \Theta(g(n))$	$f(n) = o(g(n))$	$f(n) = \omega(g(n))$
Не	Да	Не	Не	Да

□

2) Докажете или опровергайте, че ако $f_1(x) = \Omega(g_1(x))$ и $f_2(x) = \Omega(g_2(x))$, то $f_1(x).f_2(x) = \Omega(g_1(x).g_2(x))$. Функциите $f_1(x)$, $f_2(x)$, $g_1(x)$ и $g_2(x)$ са асимптотично положителни.

Решение:
Твърдението е вярно. Щом четирите функции са асимптотично положително, за всяка има стойност на аргумента, такава че за всички стойности на аргумента по-големи или равни на нея, функцията е положителна. Нека x_0 е максимумът от тези четири стойности за x.

По определението на нотацията Ω, съществува стойност на x, да я наречем x_1, и положителна константа c_1, такива че за $\forall x \geq x_1$, $f_1(x) \geq c_1.g_1(x)$. Аналогично, съществува стойност на x, да я наречем x_2, и положителна константа c_2, такива че за $\forall x \geq x_2$, $f_2(x) \geq c_2.g_2(x)$. Нека $x' = \max\{x_0, x_1, x_2\}$. Очевидно е, че $\forall x \geq x'$, $f_1(x).f_2(x) \geq c_1.c_2.g_1(x).g_2(x)$.

И така, посочихме стойност на аргумента, а именно x', и положителна константа, а именно $c_1.c_2$, за които е изпълнено $\forall x \geq x'$, $f_1(x).f_2(x) \geq c_1.c_2.g_1(x).g_2(x)$. Съгласно определението на нотацията Ω, $f_1(x).f_2(x) = \Omega(g_1(x).g_2(x))$, $\square$

3) Докажете или опровергайте, че ако $f_1(x) = \Omega(g_1(x))$ и $f_2(x) = \Omega(g_2(x))$, то $f_1(x) - f_2(x) = \Omega(g_1(x) - g_2(x))$. Функциите $f_1(x)$, $f_2(x)$, $g_1(x)$, $g_2(x)$, $f_1(x) - f_2(x)$ и $g_1(x) - g_2(x)$ са асимптотично положителни.

Решение:
Твърдението не е вярно. За да го опровергаем е достатъчно да посочим контрапример, а за да намерим подходящ контрапример е достатъчно да се посочим функции $f_1(x)$, $f_2(x)$, $g_1(x)$ и $g_2(x)$, всяка от които е сума от един и същи „водещ“ член, примерно x^2, и остатък, който е положителна или поне неотрицателна функция, но по-бавно нараства от x^2. За да са подходящи за нашето доказателство, остатъците трябва да са такива, че при изваждането, остатъкът на g_1 минус остатъка на g_2 да е по-бързо растяща функция от остатъкът на f_1 минус остатъка на f_2. Примерно, нека

$$
\begin{aligned}
f_1(x) &= x^2 + \sqrt{x} \\
f_2(x) &= x^2 \\
g_1(x) &= x^2 + x \\
g_2(x) &= x^2
\end{aligned}
$$

Очевидно четирите функции са асимптотично положителни. Функциите $f_1(x) - f_2(x) = \sqrt{x}$ и $g_1(x) - g_2(x) = x$ също са асимптотично положителни. Изпълнено е и $f_1(x) = \Omega(g_1(x))$ и $f_2(x) = \Omega(g_2(x))$. Но $f_1(x) - f_2(x) \neq \Omega(g_1(x) - g_2(x))$, тъй като $\sqrt{x} \neq \Omega(x)$. $\square$

II. Решете следните три рекурентни отношения.

1) $T(n) = 16T\left(\frac{n}{4}\right) + n^2(\lg n)^3$
2) $T(n) = 16T\left(\frac{n}{4}\right) + (\lg n)^3$
3) $T(n) = 2T(n-1) - T(n-2) + n^3 + 2^n$

Решение на 1):
Задачата не може да се реши с MT (Master Theorem) във вида, в който тя (MT) е дадена в учебника на Cormen, Leiserson, Rivest. Ако използваме терминологията на MT, $a = 16$, $b = 4$, следователно $\log_b a = 2$, следователно $n^{\log_b a} = n^2$. Функцията $f(n) = n^2(\lg n)^3$ е такава, че $f(n) \neq \Theta(n^2)$, $f(n) \neq O(n^{2-\epsilon})$ за кое да е положително ϵ, и също така $f(n) \neq \Omega(n^{2+\epsilon})$ за кое да е положително ϵ, следователно не можем да класифицираме рекурсията в нито един от трите случаи на MT. Но можем да приложим разширението на MT, което третира случаите, в които $f(n) = \Theta(n^{\log_b a} \lg^k n)$ за някаква положителна константа k – решението за тези случаи е $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$. Това разширение на MT е доказвано и на упражнения, и в сборника със задачи под името Theorem 2 на стр. 77. Изполвайки го, получаваме, че решението на текущата задача е $T(n) = n^2(\lg n)^4$.

Решение на 2):

МТ е приложима. Тъй като $(\lg n)^3 = O(n^{(\lg_4 16)-\epsilon})$ за някакво положително ϵ, примерно $\epsilon = 1$, рекурсията попада в случай 1 на МТ и решението е $T(n) = \Theta(n^{\lg_4 16})$, тоест $T(n) = \Theta(n^2)$.

Решение на 3):
Решаваме чрез метода с характеристичното уравнение. Характеристичното уравнение е

$$x^2 - 2x + 1 = 0$$

с корени $x_1 = 1$ и $x_2 = 1$. Нехомогенната част е $n^3 + 2^n = n^3.1^n + n^0.2^n$. От нея имаме два корена: 1 с кратност 4 и 2 с кратност 1 Общо за хомогенната и нехомогенната част имаме корен 1 с кратност 6 и корен 2 с кратност 1. Решението е

$$T(n) = A.1^n + B.n.1^n + C.n^2.1^n + D.n^3.1^n + E.n^4.1^n + F.n^5.1^n + G.2^n$$

за някакви положителни константи A, B, C, D, E, F и G. Ясно е, че

$$T(n) = \Theta(2^n)$$

III. Намерете асимптотичната времева сложност на алгоритмите, представени чрез следните фрагменти, като функция на n.

1)
```
int f(int n, int m) {
 int i, s = 0;
 if (n == 0 || n == 1)
  return m;
 for(i = 0; i < 5; i++)
  {
  s += f(n-1, m + i);
  s += f(n-2, m + 2*i);
  }
 s += f(n-2, 2*m)*3;
 return s; }
```

2)
```
int r(int n) {
 int i, s = 2;
 if (n == 1)
  return 2;
 for(i = n; i > 0; i /= 2)
  s += 2;
 s += r(n/2)*r(n/2);
 return s; }
```

3)
```
int p(int n) {
 int i, j, k, s = 0;
 for(i = n/2; i > 0; i--)
  for(j = 0; j < i; j++)
   for(k = j; k < i; k++)
    s ++;
 return s; }
```

Решение на 1):
Рекурентната зависимост, описваща асимптотичната времева сложност, е

$$T(n) = 5T(n-1) + 6T(n-2) + 1$$

Това е така, понеже променливата, от която зависят рекурсивните викания, е n. Другата променлива m няма отношение към бързодействието (макар че със сигурност има отношение към върнатия резултат). Забележете, че имаме шест викания с $n-2$ и пет с $n-1$.

Решението на рекурентната зависимост е чрез метода на характеристичното уравнение. В случая то е

$$x^2 - 5x - 6 = 0$$

с корени $x_1 = \frac{5+\sqrt{5^2+24}}{2} = \frac{5+\sqrt{49}}{2} = \frac{5+7}{2} = 6$ и $x_2 = \frac{5-7}{2} = -1$. От нехомогенната част имаме още един корен 1 с кратност 1. Решението е

$$T(n) = A.6^n + B.(-1)^n + C.1^n$$

за някакви положителни константи A, B и C. Ясно е, че

$$T(n) = \Theta(6^n)$$

Решение на 2):
Тъй като алгоритъмът е рекурсивен, търсим рекурентна зависимост, описваща времевата сложност. Забележете, че `for` цикълът има времева сложност $\Theta(\lg n)$ и след това се правят две рекурсивни извиквания, всяко върху вход, наполовина по-малък (игнорираме това дали n е четно или не и не се занимаваме с $\lfloor\ \rfloor$). Рекурентната зависимост, описваща асимптотичната времева сложност, е

$$T(n) = 2T\left(\frac{n}{2}\right) + \lg n$$

Тя се решава с МТ. Тъй като $\lg n = O\left(n^{(\log_2 2)-\epsilon}\right)$ за някое положително ϵ, примерно $\epsilon = \frac{1}{2}$, рекурсията попада в случай 1 на МТ и решението е $T(n) = \Theta\left(n^{\log_2 2}\right)$, тоест $T(n) = \Theta(n)$.

Решение на 3):

$$\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} \sum_{j=0}^{i-1} \sum_{k=j}^{i-1} 1 = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} \sum_{j=0}^{i-1} (i - j) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} \left(\underbrace{\sum_{j=0}^{i-1} i}_{i(i-1)} - \underbrace{\sum_{j=0}^{i-1} j}_{\frac{i(i-1)}{2}} \right) = \frac{1}{2} \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} (i^2 - i) = \Theta(n^3)$$