Семестриална контролна работа по ДАА, 07.04.2012 г.

Име: .... ФN°: .... Група: ....

Задача	1	2	3	4	5	6	7 (бонус)	Общо
получени точки								
от максимално								

Зад. 1 Нека $f(n)$, $g(n)$ и $h(n)$ са произволни асимптотично положителни функции, такива че $f(n) = \Theta(g(n))$ и $h(n) = \Theta(g(n))$. Докажете или опровергайте всяко от следните твърдения:

1. $\sqrt{f(n)h(n)} = \Theta(g(n))$
2. $\sqrt[3]{(f(n))^2 + (h(n))^2} = \Theta(g(n))$

Зад. 2 Подредете по асимптотично нарастване следните функции. Обосновете отговорите си кратко. Напишете в явен вид самата подредба.

$n!$, $100n!$, $n^2 2^n + n!$, $(n!)^2$,

$\sum_{k=1}^{n} k!$, $n^2 2^n$, n^{12}, $\sqrt[n]{n!}$

Зад. 3 Разгледайте следните два програмни фрагмента, написани на C. Докажете чрез инварианти, че всяка от функциите `sum1( )` и `sum2( )` връща сумата на елементите на масива $A[0, 1, \ldots, n-1]$:

```
int A[n];
int sum1(int n) {
   int i, s = 0;
   for(i = 0; i < n; i ++) {
         s += A[i]; }
   return s; }
```

```
int A[n];
int sum2(int n) {
   int i, s = 0;
   for(i = 0; i < n; i ++) {
      if (i%2 == 0) {
         s += A[ i/2 ]; }
      else {
         s += A[n - 1 - i/2]; } }
   return s; }
```

Имайте предвид, че целочисленото деление `i/2` връща $\left\lfloor \frac{i}{2} \right\rfloor$.

Зад. 4 Решете чрез развиване (итерация) следното рекурентно отношение

$$T(n) = T\left(\frac{n}{4}\right) + n^2$$

Зад. 5 Докажете по индукция, че следното рекурентното отношение има решение $T(n) = \Theta\left(\left(\frac{3}{2}\right)^n\right)$:

$$T(n) = \frac{3}{2}T(n-1) + 2$$

Зад. 6 Даден е пет елементен масив $A[]$, такъв че $\forall i, 1 \leq i \leq 5 : A[i] \in \{1, 2, 3, 4, 5\}$. Известно е, че елементите на $A[]$ са два по два различни. Кои са възможните стойности на $A[]$, за които изучаваната функция Build Heap($A[]$) превръща масива в $[5, 3, 4, 1, 2]$? *Пояснение:* Решаването на тази задача с "брутална сила", тоест систематичното разглеждане на всички входове, които са възможни съгласно първото и второто изречение, **1)** е много трудоемко и **2)** не носи максимален брой точки.

Зад. 7 (бонус) Помислете за идеята за троична пирамида. *Попълнено троично дърво* се дефинира аналогично на попълнено двоично дърво, само че в троичното, всеки връх има най-много три, а не два, сина. Предложете подходяща дефиниция на *максимална троична пирамида*, предложете анализи на формулите за Parent(), Left() и Right() за троична пирамида, предложете имплементации (на псевдокод) на функциите Heapify() и Build Heap(), работещи върху троична пирамида. Докажете линейна горна граница върху сложността по време на Build Heap(), когато пирамидата е троична.