Контролно по ДАА – 02.04.2013г. - решения

1 задача: Докажете или опровергайте следните две твърдения:

$$\frac{1}{n^2} \prec \frac{1}{n} \qquad\qquad 2^{\frac{1}{n^2}} \prec 2^{\frac{1}{n}}$$

Решение:

$$\lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{n^2}} = \lim_{n\to\infty} \frac{1}{n} = 0 \quad \Rightarrow \; \frac{1}{n^2} \prec \frac{1}{n}$$

$$\lim_{n\to\infty} \frac{2^{\frac{1}{n^2}}}{2^{\frac{1}{n}}} = \lim_{n\to\infty} 2^{\left(\frac{1}{n^2}-\frac{1}{n}\right)} = 1 \quad \Rightarrow 2^{\frac{1}{n^2}} \asymp 2^{\frac{1}{n}}$$

2 задача: Докажете чрез инварианта, че функцията **average** намира средното аритметично на числата в масива **A[0, 1,…,n-1]**. Допуснете, че $\boldsymbol{n}$ е поне 1 и числата в масива са цели.

```
int a[n];
double average(int n){
        int i, sum = 0, count = 0;
        for(i = 0; i < n; i++) {
                sum += a[i];
                count++;
        }
        return sum/count;
}
```

Решение: Следното твърдение е инварианта на цикъла: При всяко достигане на цикъла for на ред (4) променливата sum съдържа сумата на първите i елементи от масива, а count = i.
База: При първото достигане на ред (4) i = 0, sum = 0, count = 0 (от присвояванията на ред (3)). Очевидно е изпълнено твърдението.
Поддръжка: Да допуснем, че инвариантата е изпълнена при някое достигане на ред (4), което не е последно. Тогава в sum имаме елементите на първите i елементи на масива и count = i. В тялото на for цикъла добавяме към sum още един елемент и count и i се увеличават с 1. Тогава, спрямо новата стойност на i отново ще имаме, че count = i, а в sum ще сме добавили следващия елемент, т.е. отново в sum ще е сумата на първите i елемента.
Терминация: При последното достигане на ред (4) i = n. Т.е. променливата sum съдържа сумата на всички n елемента на масива, а count = n. Следователно на последния ред функцията average ще върне точно средното аритметично на числата в масива.

3 задача: Подредете функциите по асимптотично нарастване. Обосновете отговора си. Трябва да е абсолютно ясно и недвусмислено каква наредба сте намерили.

$$(n!)^n \qquad 2^{2^n} \qquad (n!)^2 \qquad \sum_{k=1}^{n!} k^2 \qquad n^{n!}$$

Решение:

$$\sum_{k=1}^{n!} k^2 = \frac{(n!)(n!+1)(2n!+1)}{6} \asymp (n!)^3 \succ (n!)^2$$

От друга страна

$$(n!)^n \succ (n!)^3 \asymp \sum_{k=1}^{n!} k^2$$

$2^{2^n} \succ (n!)^n$, защото след логаритмуване получаваме $2^n \succ nlg(n!) \asymp n^2 lgn$, което очевидно е вярно.

$n^{n!} \succ 2^{2^n}$, защото след логаритмуване получаваме $n!\, lgn \succ 2^n$, което очевидно е вярно.

Търсената подредба е:

$$n^{n!} \succ 2^{2^n} \succ (n!)^n \succ \sum_{k=1}^{n!} k^2 \succ (n!)^2$$

4 задача: Намерете асимптотичната сложност по време на програмните фрагменти. В първия фрагмент ще получите 4 бонус точки, ако изведете и точния израз, който връща функцията **f**, като функция на $\boldsymbol{n}$.

```
int f(int n) {
        int i, j, s = 0;
        for(i = 0; i <= n; i++)
                for(j = 0; j <= i; j++)
                        s++;
        return s;
}
```

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

Решение: За функцията **f** имаме, че вътрешният цикъл се изпълнява по i+1 пъти за i от 0 до n ⇒ $s = \sum_{j=1}^{n+1} j = \frac{(n+1)(n+2)}{2}$ и сложността по време е $\theta(n^2)$.
За функцията **func** имаме, че вътрешният цикъл се изпълнява по i пъти ⇒

$$n + \frac{n}{2} + \frac{n}{4} + \cdots = n\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) \asymp n\frac{1}{1-\frac{1}{2}} = \theta(n)$$