1.test DSA 6.4.2018

1. Příklad na dobu výpočtu

Který fragment programu z následujících dvou proběhne rychleji? Kolikrát bude v každém z řešení proveden příkaz sum += i+j, tj. tělo cyklu?

```
int n = 100;
int sum = 0;
for (i = 0; i < n; i++)
      for (j = 0; j < i; j++)
            sum += i+j;

int n = 75;
int sum = 0;
for (i = 0; i < n; i++)
      for (j = 0; j < n; j++)
            sum += i+j;
```

Řešení:

V prvním fragmentu proběhne vnější cyklus 100 kráte, vnitřní však proběhne nejprve 0 krát, pak jednou, dvakrát a nakonec 99 krát. Tělo vnitřního cyklu se tedy provede

0 + 1 + 2 + ... + 99 = (100/2)·(0+99) = 50 · 99 = 4950 krát.

Ve druhém fragmentu není počet provádění vnitřního cyklu závislý na proměnné vnějšího cyklu, vnější cyklus proběhne 75 krát a v jeho těle pokaždé proběhne vnitřní cyklus rovněž 75 krát. Celkem se tedy tělo vnitřního cyklu provede 75 *75 = 5625 krát.

Rychleji proběhne první fragment.

2. Příklad na rekurzi

Napište rekurzivní funkci f12, která pro zadané číslo N vypíše řetězec skládající se z N jedniček následovaných 2*N dvojkami. Např. pro N = 3 vypíše 111222222.

Řešení:

```
void f12 (int n) {
      if (n <= 0) return;
      printf("1");
      f12(n-1);
      printf("22");
}
```

3. Příklad na asymptotickou složitost

Dokažte, že pro libovolné reálné konstanty a, b, kde b > 0, platí: $(n + a)^b = \Theta(n^b)$.

Řešení:

Abychom dokázali $(n + a)^b = \Theta(n^b)$, musíme najít c_1, c_2 a n_0 takové, že $0 \le c_1 n^b \le (n + a)^b \le c_2 n^b$ pro $n \ge n_0$.

Pro $n \ge |a|$ platí: $n + a \le n + |a| \le 2n$.

Pro $n/2 \ge |a|$ platí: $n + a \ge n - |a| \ge n/2$.

Pro $n \ge 2|a|$ platí: $0 \le n/2 \le n + a \le 2n$.

Protože b>0, platí totéž i po umocnění na b:

$0^b \le (n/2)^b \le (n + a)^b \le (2n)^b$
$0 \le (1/2)^b n^b \le (n + a)^b \le 2^b n^b$

Zvolíme-li $c_1 = (1/2)^b$, $c_2 = 2^b$ a $n_0 = 2|a|$, pak $0 \le c_1 n^b \le (n + a)^b \le c_2 n^b$ platí.

4. Příklad na invarianty

Určete a vyznačte tu část kódu, která porušuje daný invariant.

```
int f(int[] a) { // vzdy plati a.length > 0
        int r = a[0];
        int i = a.length;
        do {
                i--;
                r = r + a[i];
        // invariant: v r je soucet prvku na indexech i..a.length - 1
        } while (i > 0);
        return 0;
}
```

Řešení:

Invariant porušuje druhá řádka, měla by zajistit, aby při první iteraci platilo **r == a[a.length - 1]**, tzn. mělo by na ní být **int r = 0**.

5. Příklad na asymptotickou složitost

Zjistěte asymptotickou složitost:

$T(1) = 1$
$T(n) = 2T(n/2) + n^2$

mistrovskou metodou (pomocí Master teorému). Pečlivě zkontrolujte a zapište všechny předpoklady věty!

Řešení:

Abychom mohli použít master teorém musíme porovnat $n^{\lg 2}$ (tj. n) a n^2 (a=2, b=2). Zřejmě $f(n) = n^2 = \Omega(n^{\lg 2+e})$, kde $e = 1 > 0$. Dále musí platit $2 \cdot f(n/2) \le c \cdot f(n)$ pro dostatečně veliká n. Tj. $2 \cdot (n/2)^2 = n^2/2 \le c \cdot n^2$ a to platí např. pro c=1/2. Platí tedy třetí možnost teorému a $T(n) = \Theta(n^2)$.