Cvičení 8

Příklad 1: Průměr z daných $n > 1$ čísel spočítáme následující funkcí:

```
PRUMER(X, n)
z ← 0.0
for i ← 1 to n
        do z ← z + X[i]
return z/n
```

Určete, kolik tato funkce `prumer` vykoná aritmetických operací v závislosti na n.

Příklad 2: Určete, kolik průchodů vnitřním cyklem provede pro vstup n následující jednoduchý program.

```
ALG1(n)
for i ← 1 to n ∗ n
        do for j ← 1 to i
                do print "jeden průchod"
```

Je to v asymptotické notaci $\Theta(n^2)$ nebo $\Theta(n^3)$ nebo $\Theta(n^4)$?

Příklad 3: Jednoduchá implementace Euklidova algoritmu největšího společného dělitele dvou přirozených čísel a, b počítá výsledek následovně:

```
EUCLID(a, b)
if b = 0
    then return a
    else if a > b
            then return EUCLID(b, a − b)
            else return EUCLID(a, b − a)
```

Nechť k označuje počet bitů v binárním zápise čísel a, b. Jaká je nejhorší možná časová složitost zadaného algoritmu vzhledem ke k? (Uvažujte jednotkový čas na každou aritmetickou operaci.)

Příklad 4: Kolik kroků (elementárních výpočetních operací) provede následující efektivní implementace Euklidova algoritmu pro největšího společného dělitele na vstupech – číslech a, b, která mají v binárním zápise nejvýše ℓ bitů? (Předpokládejte, že aritmetické operace trvají jednotkový čas.)

```
EUCLID(a, b)
while b ≠ 0
        do c ← a mod b
           a ← b
           b ← c
return a
```

Příklad 5: Představme si, že z daných n čísel máme vybrat k-té v uspořádání podle velikosti. (Nejpřirozenějším postupem by bylo čísla setřídit a pak k-té z nich vybrat, ale to je spousta zbytečných výpočtů navíc, že?)

Pro rychlý výpočet použijeme následující rekurzivní algoritmus, svým způsobem podobný algoritmu quicksort.

Rekurzivní procedura $\textsc{Vyber}(A, p, r, k)$ vracející k-tý nejmenší prvek z úseku pole $A[p\mathinner{..}r]$ (kde $p \leq k \leq r$) pracuje následovně:

- Z úseku $A[p\mathinner{..}r]$ zvolíme libovolný prvek x a tento úsek přeuspořádáme tak, že bude rozdělen na tři části:
  - $A[p\mathinner{..}q-1]$ – obsahující prvky, které jsou menší než x,
  - $A[q\mathinner{..}q']$ – obsahující prvky, kreré jsou rovny x,
  - $A[q'+1\mathinner{..}r]$ – obsahující prvky, které jsou větší než x
- Pokud je $q \leq k \leq q'$, je výsledkem x.
- Pokud $k < q$, výsledek spočítáme rekurzivním výpočtem jako $\textsc{Vyber}(A, p, q-1, k)$.
- Pokud $k > q'$, výsledek spočítáme rekurzivním výpočtem jako $\textsc{Vyber}(A, q'+1, r, k)$.

Jaká je časová složitost popsaného algoritmu?

Příklad 6: Určete, kolik průchodů vnitřním cyklem provede pro vstup n následující jednoduchý program.

$\textsc{Alg2}(n)$

for $i \leftarrow 1$ **to** n
    do for $j \leftarrow 1$ **to** $i * i$
        do print "`jeden průchod`"

Je to v asymptotické notaci $\Theta(n^2)$ nebo $\Theta(n^3)$ nebo $\Theta(n^4)$?

Příklad 7: Určete, kolik průchodů cyklem provede následující program pro vstup n. Zapište výsledek v asymptotické notaci $\Theta(\cdot)$.

$\textsc{Alg3}(n)$

$i \leftarrow 1$
while $i < n$
    do print "`jeden průchod`"
        $i \leftarrow i + i$

***Příklad 8:** Určete, kolik průchodů cyklem provede následující program pro vstup n. Zapište výsledek v asymptotické notaci $\Theta(\cdot)$.

Alg4(n)

```
i ← 1; j ← 1
while i < n
      do print "jeden průchod"
         i ← i + j
         j ← j + 1
```

***Příklad 9:** Jak byste upravili algoritmus v Příkladě 5, aby počítal v (nejhorším) lineárním čase?

Návod: Je potřeba najít vhodný způsob výběru prvku x tak, aby bylo zaručeno, že rozdělení nebude velmi nevyvážené.

Příklad 10: Rozhodněte, které z následujících asymptotických vztahů mezi funkcemi proměnné n jsou platné. (Pozor, platné mohou být oba nebo i žádný.)

a) $\log n \in O(\sqrt{n})$

b) $\sqrt{n} \in O(n)$

Příklad 11: Rozhodněte, které z následujících asymptotických vztahů mezi funkcemi proměnné n jsou platné. (Pozor, platné mohou být oba nebo i žádný.)

a) $2^n \in O(n^n)$

b) $2^n \in O(n^{1024})$

Příklad 12: Rozhodněte, které z následujících asymptotických vztahů mezi funkcemi proměnné n jsou platné. (Pozor, platné mohou být oba nebo i žádný.)

a) $n! \in O(2^n)$

b) $n^{\log n} \in O(n^{1024})$