Příklad 1/23

Pro rostoucí spojité fukce $f(x)$, $g(x)$ platí $f(x) \in \Omega(g(x))$.
Z toho plyne, že:

a) $f(x) \in O(g(x))$

b) $f(x) \in \Theta(g(x))$

c) $g(x) \in \Theta(f(x))$

d) $g(x) \in \Omega(f(x))$

e) $g(x) \in O(f(x))$

Příklad 2/23

Pro rostoucí spojité fukce $f(x)$, $g(x)$ platí $f(x) \in O(g(x))$.
Z toho plyne, že:

a) $f(x) \in \Theta(g(x))$

b) $f(x) \in \Omega(g(x))$

c) $g(x) \in \Theta(f(x))$

d) $g(x) \in \Omega(f(x))$

e) $g(x) \in O(f(x))$

Příklad 3/23

Pokud funkce f roste asymptoticky rychleji než funkce g (tj. $f(x) \notin O(g(x))$), platí následující tvrzení:

a) jsou-li v bodě x definovány obě funkce, pak $f(x) > g(x)$

b) rozdíl $f(x) - g(x)$ je vždy kladný

c) rozdíl $f(x) - g(x)$ je kladný pro každé $x > y$,

kde y je nějaké dostatečně velké číslo

d) obě funkce f i g jsou definovány jen pro nezáporné argumenty

e) nic z předchozího

Příklad 4/23

Pokud funkce f roste asymptoticky stejně rychle jako funkce g
(tj. $f(x) \in \Theta(g(x))$), platí právě jedno následující tvrzení. Které?

a) jsou-li v bodě x definovány obě funkce, pak $f(x) = g(x)$

b) ani poměr $f(x)/g(x)$ ani poměr $g(x)/f(x)$ nekonverguje k nule s rostoucím x

c) rozdíl $f(x) - g(x)$ je kladný pro každé $x > y$, kde y je nějaké dostatečně velké číslo

d) obě funkce f i g jsou definovány jen pro nezáporné argumenty

e) nic z předchozího

Příklad 5/23

Pro dvě spojité funkce f(x) a g(x) rostoucí na celém **R** platí f(x) < g(x) pro každé x $\in$ **R**. To znamená:

a) $f(x) \notin \Omega(g(x))$

b) $f(x) \notin O(g(x))$

c) je možné, že $f(x) \in \Omega(g(x))$

d) $g(x) \notin \Theta(f(x))$

e) f(x) roste asymptoticky pomaleji než g(x)

Příklad 6/23

Pro dvě spojité funkce f(x) a g(x) rostoucí na celém **R** platí $f(x) \notin \Omega(g(x))$, $f(x) \notin \Theta(g(x))$. Tudíž:

a) $g(x) \in O(f(x))$

b) $g(x) \in \Theta(f(x))$

c) $f(x) < g(x)$ pro každé $x \in$ **R**

d) $f(x) \leq g(x)$ pro každé $x \in$ **R**

e) může existovat $y \in$ **R** takové, že $f(y) > g(y)$

Příklad 7/23

Algoritmus A probírá postupně všechny prvky v dvourozměrném poli o velikosti $n \times n$ a s každým prvkem provádí další (nám neznámou) akci, jejíž složitost je $\Theta(\log_2(n))$. Celková asymptotická složitost algoritmu A je tedy:

a) $\Theta(n \cdot \log_2(n))$

b) $\Theta(n^2)$

c) $\Theta(n^3)$

d) $\Theta(n^2+\log_2(n))$

e) $\Theta(n^2 \cdot \log_2(n))$

Příklad 8/23

Právě jeden z následujících výroků je nepravdivý. Označte jej.

a) $x \cdot \log_2(x) \in O(x^2 - x)$

b) $x \cdot \log_2(x) \in O(x^2 - \log_2(x))$

c) $x \cdot \log_2(x) \in \Omega(x^2 - \log_2(x))$

d) $x \cdot \log_2(x) \in \Omega(x + \log_2(x))$

e) $x \cdot \log_2(x) \in \Theta(x \cdot \log_2(x^2))$

Příklad 9/23

Algoritmus A provede jeden průchod polem s *n* prvky. Při zpracování prvku na pozici *k* provede *k*+*n* operací. Operační (=asymptotická) složitost algoritmu A je tedy:

a) $\Theta(k+n)$

b) $\Theta(\ (k+n)\cdot n\ )$

c) $\Theta(k^2+n)$

d) $\Theta(n^2)$

e) $\Theta(n^3)$

Příklad 10/23

V následujících vztazích doplňte na prázdná místa (........) symboly O nebo Θ nebo Ω tak, aby vznikla pravdivá tvrzení. Je-li možností více, uveďte je všechny, nehodí-li se ani jeden symbol, prázdné místo proškrtněte.

a) $x^2 \cdot 2^x \in$ ....$((\ln(x^2))^2 + 2^x)$

b) $(\ln(x^2))^2 + 2^x \in$ ....$(x^2 + \ln(x^2)\ )$

c) $2^x \cdot (\ln(x))^{-1} \notin$ ....$(2^x \cdot (\ln(x^2))^{-1})$

Příklad 11/23

V následujících vztazích doplňte na prázdná místa (........) symboly O nebo Θ nebo Ω tak, aby vznikla pravdivá tvrzení. Je-li možností více, uveďte je všechny, nehodí-li se ani jeden symbol, prázdné místo proškrtněte.

a) $x^2 \cdot \ln(x^2) \in$ ....$(x^2 + \ln(x))$

b) $x^3 + \ln(x^2) \in$ ....$(x^3 + 2^x)$

c) $x^3 \cdot \ln(x^2) \notin$ ....$(\ln(x^2) + 2^x)$

Příklad 12/23

Uveďte příklad tří rostoucích funkcí reálné proměnné f(x), g(x) a h(x), pro které současně platí všechny tři následující vztahy:

$$f(x) \notin O(g(x)),\ \ g(x) \notin \Theta(h(x)),\ \ h(x) \notin \Omega(f(x))$$

Pokud taková trojice funkcí nemůže existovat, napište krátké zdůvodnění, proč.

Příklad 13/23

Uveďte příklad tří rostoucích funkcí reálné proměnné f(x), g(x) a h(x), pro které současně platí všechny tři následující vztahy:

$$f(x) \notin O(g(x)),\ \ g(x) \notin \Omega(h(x)),\ \ h(x) \notin \Theta(f(x))$$

Pokud taková trojice funkcí nemůže existovat, napište krátké zdůvodnění, proč.

Příklad 14/23

Matice *A* má *M* řádků a *N* sloupců indexovaných od 0. Na zpracování prvku matice na pozici $[r][s]$ $(0 \leq r < M,\ 0 \leq s < N)$ je zapotřebí právě *s* operací, z nichž každá má konstantní složitost. Jaká je asymptotická složitost zpracování celé matice?

Příklad 15/23

Matice A má M řádků a N sloupců indexovaných od 0. Na zpracování prvku matice na pozici $[r]\,[s]$ $(0 \le r < M,\ 0 \le s < N)$ je zapotřebí právě $s+r$ operací, z nichž každá má konstantní složitost. Jaká je asymptotická složitost zpracování celé matice?

Využite vztah:

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

Příklad 16/23

Uvažte algoritmus násobení dvou celých čísel, tak jak je znám ze školy pro ruční násobení. Předpokládejte, že sečtení nebo vynásobení dvou *číslic* má konstantní časovou složitost.

Určete asymptotickou složitost vynásobení dvou celých čísel M, N zapsaných v desítkové soustavě.

```
Příklad násobení           9803
                         x 347
M =  9803               -------
N =  347                 68621
                        39212
                       29409
                      ---------
                       3401641
```

Příklad 17/23

Určete asymptotickou složitost daného kódu v závislosti na N.

```
//a = array[0..N-1] of int;
for(i = 0; i < N; i++)
  a[i] = N;
for (i = 0; i < N; i++)
  while (a[i] > 0) {
     print(a[i]);
     a[i] = a[i]/2;    // integer division
  }
```

Příklad 18/23

Určete asymptotickou složitost daného kódu v závislosti na N.

```
//a = array[0..N-1] of int;
for(i = 0; i < N; i++)
  a[i] = i;
for (i = 0; i < N; i++)
  while (a[i] > 0) {
     print(a[i]);
     a[i] = a[i]/2;    // integer division
  }
```

Příklad 19/23

Určete asymptotickou složitost daného kódu v závislosti na N.

```
//a = array[0..N-1] of int;
for(i = 0; i < N; i++)
  a[i] = 1;
for (i = 1; i < N; i++)
  while (a[i] <= 2*a[i-1]) {
     print(a[i]);
     a[i] = a[i]+1;
  }
```

Příklad 20/23

A. Jaká je asymptotická složitost vynásobení dvou matic o velikosti N x N?

B. Jaká je asymptotická složitost Gaussova eliminačního algoritmu pro soustavu N rovnic o N neznámých?

C. Jaká je asymptotická složitost výpočtu determinantu matice velikosti N x N přímo z definice determinantu?

Lze determinant vypočítat efektivněji, s nižší asymptotickou složitostí? Jak?

D. Jaká je asymptotická složitost výpočtu řešení soustavy N lineárních rovnic s N neznámými pomocí Cramerova pravidla?

Příklad 21/23

Na obvodu kružnice jsou v libovolně nepravidelných intervalech vyznačeny body očíslované po řadě za sebou 1, 2, ..., N. Máme určit počet všech takových trojúhelníků, jejichž vrcholy leží v očíslovaných bodech a které neobsahují střed kružnice jako svůj vnitřní bod.

Navrhněte algoritmus a určete jeho asymptotickou složitost.

Řešte analogickou úlohu pro konvexní čtyřúhelníky.

Příklad 22/23

Na výstup máme vypsat všechna kladná celá čísla, která jsou menší než dané číslo N a která ve svém binárním zápisu obsahují právě 3 jedničky.

Jaký bude asymptotická složitost efektivního algoritmu?

Algoritmus lineární vůči N je neefektivní.

Příklad 23/23

Popište, jak vypočtete hodnotu

$$\log(\log(N^{N!}))$$

pro $N = 10^7$.

Jak dlouho bude trvat výpočet na Vašem osobním počítači?
Logaritmus je o základu 10.

Nepoužívejte aproximace jako např. Stirlingův vzorec apod.