

### Doba běhu daného algoritmu/programu

1.

Který fragment programu z následujících dvou proběhne rychleji?

```
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 krát, vnitřní však proběhne nejprve 0 krát, pak 1 krát, 2 krát a nakonec 99 krát. Tělo vnitřního cyklu se tedy provede  $0 + 1 + 2 + \dots + 99 = (100/2) \cdot (0+99) = 50 \cdot 99 = 4950$  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  $75 \cdot 75 = 5625$  krát. Rychleji proběhne první fragment.

2.

Do následujícího kódu doplňte chybějící konstantu v podmínce tak, aby byla procedura `xyz()` volána právě 2100 krát.

```
for (i=0; i < 70; i++) {
    j = 0;
    do {
        if (j > ____ ) xyz();
        j++;
    } while (j < 90);
}
```

Řešení

Vnější cyklus probíhá od 0 do 69, celkem 70 krát. Musíme proto zajistit, aby ve vnitřním cyklu byla procedura `xyz()` volána právě  $2100/70 = 30$  krát. Toho docílíme pro hodnoty řídicí proměnné `j`: 60, 61, ..., 88, 89. Chybějící konstanta je tedy 59.

3.

Do následujícího kódu doplňte chybějící konstantu v podmínce tak, aby byla procedura `xyz()` volána právě 2000 krát.

```
i = 50;
do {
    for (j=0; j < 70; j++)
        if (j > ____ ) xyz();
    i++;
} while (i < 150);
```

## Řešení

Vnější cyklus probíhá od 50 do 149, celkem 100 krát. Musíme proto zajistit, aby ve vnitřním cyklu byla procedura xyz() volána právě  $2000/100 = 20$  krát. Toho docílíme pro hodnoty řídicí proměnné j: 50, 51, ..., 68, 69. Chybějící konstanta je tedy 49.

## 4.

Do následujícího kódu doplňte chybějící výraz v podmínce tak, aby byla procedura uvw() volána právě 49 krát.

```
for (i = 0; i < 7; i++) {
    j = i;
    while (j < ____ ) {
        uvw();
        j++;
    }
}
```

## Řešení

Vnější cyklus probíhá celkem 7 krát. Musíme proto zajistit, aby ve vnitřním cyklu byla procedura uvw() volána právě  $49/7 = 7$  krát. Protože počítání ve vnitřním cyklu začíná pokaždé od aktuální hodnoty i, stačí do podmínky doplnit výraz  $i+7$ .

## 5.

Do následujícího kódu doplňte chybějící konstantu v podmínce tak, aby byla procedura uvw() volána právě 85 krát.

```
i = 0;
while (i < 10) {
    for (j = i; j < ____; j++)
        uvw();
    i++;
}
```

## Řešení

Označme hledanou konstantu k. Při prvním průchodu vnějšího cyklu proběhne vnitřní cyklus k krát. Při druhém průchodu vnějšího cyklu proběhne vnitřní cyklus k-1 krát. Při třetím průchodu vnějšího cyklu proběhne vnitřní cyklus k-2 krát, atd. Protože vnější cyklus proběhne celkem 10 krát, platí rovnost:

$$k + (k-1) + (k-2) + (k-3) + \dots + (k-8) + (k-9) = 85$$

Odstraněním závorek získáme :

$$10k - 1 - 2 - 3 - \dots - 8 - 9 = 85$$

$$10k - 45 = 85$$

$$10k = 130$$

$$k = 13.$$

6.

Ke zpracování  $k$ -tého řádku matice velikosti  $n \times n$  je zapotřebí  $2k$  operací. Celkem je ke zpracování matice zapotřebí operací

- a)  $2n^2$
- b)  $(n^2)/2$
- c)  $n(n+1)/2$
- d)  $n(n-1)$
- e)  $n(n+1)$

Řešení

Celkem je zapotřebí operací

$$2 \cdot 1 + 2 \cdot 2 + 2 \cdot 3 + \dots + 2 \cdot n = 2 \cdot (1 + 2 + 3 + \dots + n) = 2 \cdot n \cdot (n+1) / 2 = n \cdot (n+1).$$

Platí varianta e).

7.

Úloha, jejíž doba řešení je  $C \cdot n^2$ , kde  $n$  je rozsah vstupních dat, se řeší na počítači pro  $n = 5000$ . Je zakoupen nový počítač, který je cca 2.5 krát rychlejší. Jak je možno zvětšit rozsah vstupních dat, aby byla úloha vyřešena na novém počítači ve stejném čase?

Řešte pro různé závislosti doby řešení na rozsahu vstupních dat:

$C \cdot n^3$ ,  $C \cdot n^{0.5}$ ,  $C \cdot n \cdot \log_2(n)$  ...

Řešení

Na starém počítači se v daném čase  $T$  musí provést  $C \cdot 5000^2$  operací.

Na novém počítači se ve stejném čase  $T$  provede 2.5 krát více operací, tedy  $2.5 \cdot C \cdot 5000^2$  operací. V čase  $T$  se má také provést  $C \cdot x^2$  operací, kde  $x$  neznámý nový rozsah vstupních dat. Musí tedy platit rovnost

$$2.5 \cdot C \cdot 5000^2 = C \cdot x^2$$

odtud získáváme

$$x^2 = 2.5 \cdot 5000^2 = 2.5 \cdot 25000000 = 62500000.$$

$$x = 7905.7 \text{ přibližně.}$$

Nový počítač je sice 2.5 krát rychlejší, ale rozsah vstupních dat lze zvětšit jen  $\sqrt{2.5}$  krát, tedy cca 1.58 krát.

Platí samozřejmě také  $5000 \cdot \sqrt{2.5} = 7905.6941504\dots$

Analogicky pro další zadané závislosti doby řešení na rozsahu vstupních dat získáme:

$$2.5 \cdot C \cdot 5000^3 = C \cdot x^3 \Rightarrow 312500000000 = x^3 \Rightarrow x = 6786.04$$

Rozsah dat lze zvětšit cca  $6786.04/5000 = 1.357$  krát. ( $\text{power}(2.5, 1/3) = 1.357$ )

$$2.5 \cdot C \cdot 5000^{0.5} = C \cdot x^{0.5} \Rightarrow 176.7766952966 = x^{0.5} \Rightarrow x = 31250.$$

Rozsah dat lze zvětšit  $31250/5000 = 6.25$  krát. ( $2.5^2 = 6.25$ )

$$2.5 \cdot C \cdot 5000 \cdot \log_2(5000) = C \cdot x \cdot \log_2(x) \Rightarrow 153596.4047 = x \cdot \log_2(x) \Rightarrow x = 11397.41931$$

Rozsah dat lze zvětšit cca  $11397.41931/5000 = 2.28$  krát.

8.

Stroj provádí  $10^9$  operací za sekundu. Pro výpočet je k dispozici 1 hodina. Určete, jaká může být maximální hodnota  $n$ , která určuje velikost vstupních dat, v případě že počet nezbytných instrukcí pro zpracování dat o velikosti  $n$  je:  $n^{3/2}$ ,  $n^{5/4}$ ,  $n \cdot \log_2(n) \cdot \log_2(\log_2(n))$ ,  $n^2 \cdot \log_2(n)$ , ... a další.

Řešení

Jedna hodina představuje  $10^9 \cdot 3600 = 3.6 \cdot 10^{12}$  operací. Snadno dopočítáme:

$$n^{3/2} \leq 3.6 \cdot 10^{12} \Rightarrow n = \lfloor (3.6 \cdot 10^{12})^{2/3} \rfloor = \lfloor 234892058.470132 \rfloor = 234\,892\,058$$

$$n^{5/4} \leq 3.6 \cdot 10^{12} \Rightarrow n = \lfloor (3.6 \cdot 10^{12})^{4/5} \rfloor = \lfloor 11\,092\,820\,887.3183 \rfloor = 11\,092\,820\,887$$

$$n \cdot \log_2(n) \cdot \log_2(\log_2(n)) \leq 3.6 \cdot 10^{12} \Rightarrow n = \lfloor 2.0608019582628 \cdot 10^{10} \rfloor = 20\,608\,019\,582$$

$$n^2 \cdot \log_2(n) \leq 3.6 \cdot 10^{12} \Rightarrow n = \lfloor 438277.5717 \rfloor = 438\,277$$

(K vypočtení posledních dvou hodnot použijeme vhodný numerický software :-))

2.06080195826280

9.

Metoda A potřebuje k vyřešení úlohy  $n^2 + 17$  operací, Metoda B potřebuje  $2n + 80$  operací, přičemž celé číslo  $n$  popisuje rozsah vstupních dat. Pro jaká  $n$  je výhodnější použít metodu A?

Řešení

Metoda A je výhodnější pro taková  $n$ , pro která platí  $n^2 + 17 < 2n + 80$ , to jest, pro taková  $n$ , která splňují nerovnici  $n^2 - 2n - 63 < 0$ .

Kořeny příslušné kvadratické rovnice  $n^2 - 2n - 63 = 0$

jsou  $n_1 = -7$ ,  $n_2 = 9$ .

Pro  $n$  z intervalu  $(-7, 9)$  má výraz  $n^2 - 2n - 63$  zápornou hodnotu, a proto je metoda A výhodnější pro  $n < 9$ .

### Asymptotická složitost

10.

Pro rostoucí spojitě 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)  $g(x) \in O(f(x))$

Řešení

$f(x) \in \Omega(g(x))$  znamená, že  $f(x)$  roste rychleji nebo stejně rychle jako  $g(x)$ . Tudíž  $g(x)$  roste pomaleji nebo stejně rychle jako  $f(x)$ . To vyjadřujeme zápisem  $g(x) \in O(f(x))$ . Platí možnost e).

11.

Pro rostoucí spojitě 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)  $g(x) \in O(f(x))$

Řešení

$f(x) \in O(g(x))$  znamená, že  $f(x)$  roste pomaleji nebo stejně rychle jako  $g(x)$ . Tudíž  $g(x)$  roste rychleji nebo stejně rychle jako  $f(x)$ . To vyjadřujeme zápisem  $g(x) \in \Omega(f(x))$ . Platí možnost d).

12.

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

Řešení

Řeč je o asymptotické složitosti, takže vztah funkcí  $f(x)$  a  $g(x)$  blízko nuly není nijak podstatný. Varianty a) b) a d) žádají z tohoto hlediska příliš mnoho a tedy neplatí.

Tvrzení úlohy  $f(x) \notin O(g(x))$  říká, že od určitého okamžiku musí platit  $f(x) > g(x)$ , což je

ekvivalentní variantě c).

**13.**

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

**Řešení**

Stejně rychlý asymptotický růst neznámá, že se musí funkce nutně rovnat, varianta a) odpadá. Stejně tak neznámá, že by jedna funkce musela nutně být větší než druhá, odpadá i varianta c). Chování funkcí v okolí nuly není pro asymptotické chování významné, varianta d) zbytečně žádá příliš mnoho. Stejně rychlý asymptotický růst znamená, že v okolí nekonečna poměr  $f(x)/g(x)$  ani poměr  $g(x)/f(x)$  nepřekročí nějakou pevnou konstantu. To ovšem znamená, že žádný z těchto poměrů nekonverguje k 0, platí varianta b).

**14.**

Pro dvě spojitě funkce  $f(x)$  a  $g(x)$  rostoucí na celém  $\mathbf{R}$  platí  $f(x) < g(x)$  pro každé  $x \in \mathbf{R}$ . 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  $g(x)$

**Řešení**

Pokud je  $f(x) < g(x)$ , nevíme mnoho co se týče asymptotické složitosti těchto funkcí.  $f(x)$  může být vůči  $g(x)$  nepatrná, maličká, a pak by *mohla* být  $f(x)$  asymptoticky pomalejší.

Taky by ale mohla být  $f(x)$  docela srovnatelná s  $g(x)$ , např.  $f(x) = g(x) - 0.01$  všude, to by pak rostly obě asymptoticky stejně rychle, tj. *mohli* bychom říct, že  $f(x) \in \Theta(g(x))$ .

Platí ale:  $f(x) \in \Theta(g(x)) \Rightarrow f(x) \in O(g(x)) \ \& \ f(x) \in \Omega(g(x))$ .

První dvě možnosti odpovědí popírají právě uvedené zjištění. Popírá ji i poslední, protože říká totéž co první (souhlasíte?). Čtvrtá možnost říká, že *určitě* „ $g$  není výrazně nad  $f$ “, ale proč by to tak nemohlo být, když  $f$  je podle zadání menší?

15.

Pro dvě spojitě funkce  $f(x)$  a  $g(x)$  rostoucí na celém  $\mathbf{R}$  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  $x \in \mathbf{R}$
- d)  $f(x) \leq g(x)$  pro každé  $x \in \mathbf{R}$
- e) může existovat  $y \in \mathbf{R}$  takové, že  $f(y) > g(y)$

Řešení

Zadání praví, že  $f(x) \notin \Omega(g(x))$ ,  $f(x) \notin \Theta(g(x))$ . Takže platí  $f(x) \in O(g(x))$  (pročpak?). Ekvivalentně můžeme říci, že  $g(x) \in \Omega(f(x))$ . Z posledního ovšem nikterak neplyne ani  $g(x) \in O(f(x))$ , ani  $g(x) \in \Theta(f(x))$  (malujte si obrázek, je-li to zapotřebí), tudíž první dvě varianty odpovědí neplatí. Hovoříme o asymptotickém chování funkcí, tedy jejich vzájemné vztahy na nějaké omezené podmnožině  $\mathbf{R}$  (třeba kolem 0), mohou být zcela libovolné, tvrdit třetí nebo čtvrtou možnost je tu opovázlivostí příliš velkou. To ostatně potvrzuje i správná pátá možnost.

16.

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

Řešení

Je nutno provést  $n \times n$  akcí o složitosti úměrné  $\log_2(n)$ , tedy složitost všech akcí dohromady je úměrná  $n^2 \cdot \log_2(n)$ . Přesun od jednoho prvku k druhému v poli má složitost konstantní, takže samotný průchod celým polem má složitost úměrnou konstanta  $\cdot n^2$ . Celkem tak máme složitost  $\Theta(n^2 \cdot \log_2(n) + \text{konstanta} \cdot n^2) = \Theta(n^2 \cdot \log_2(n))$ . To je právě varianta e).

17.

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

**Řešení**

Když k funkci  $f$  přičteme funkci  $g$ , která roste asymptoticky pomaleji, než funkce  $f$ , výsledný součet poroste asymptoticky stejně rychle jako původní funkce  $f$ . Proto můžeme pravou stranu zadání v případě a) – d) zjednodušit:

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

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

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

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

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

Funkce  $\log_2(x)$  roste pomaleji než funkce  $x$ , takže funkce  $x \cdot \log_2(x)$  roste pomaleji než funkce  $x \cdot x = x^2$ , takže varianta a) i b) je pravdivá. Funkce  $\log_2(x)$  má v plus nekonečnu limitu plus nekonečno, takže funkce  $x \cdot \log_2(x)$  roste asymptoticky rychleji než funkce  $x$ , takže varianta d) je pravdivá. Varianta e) po naznačené úpravě je také zřejmě pravdivá. Zbývá jediná nepravdivá varianta c), jejíž nepravdivost ostatně vyplývá již z první věty tohoto odstavce.

**18.**

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)  $\Theta(n^3)$

**Řešení**

Celkem je provedený počet operací při zpracování všech prvků roven

$$(1+n) + (2+n) + (3+n) + \dots + (n+n) =$$

$$= n \cdot ((1+n) + (n+n))/2 = n \cdot (1+3n)/2 = 3n^2/2 + n/2 \in \Theta(n^2)$$

Povážíme-li navíc ještě režii na přesun od jednoho prvku pole k následujícímu, která má nanejvýš konstantní složitost, připočteme k výsledku ještě člen  $\Theta(konst \cdot n)$ , jenž ovšem na složitosti  $\Theta(n^2)$  nic nemění. Platí varianta d).

**19.**

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

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

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

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



Řešení

a) funkce nalevo zřejmě roste rychleji než funkce napravo, neboť obě funkce obsahují výraz  $2^x$ , který je však nalevo vynásoben výrazem  $x^2$ , jenž roste asymptoticky rychleji než (pouze!) aditivní výraz  $(\ln(x^2))^2$  napravo. Jediná správná možnost je proto  $x^2 \cdot 2^x \in \Omega((\ln(x^2))^2 + 2^x)$ .

b) funkce nalevo obsahuje člen  $2^x$ , funkce napravo je součtem dvou členů, které oba zřejmě rostou asymptoticky pomaleji než  $2^x$ . Funkce nalevo tedy roste asymptoticky rychleji, takže jediná správná možnost je  $(\ln(x^2))^2 + 2^x \in \Omega(x^2 + \ln(x^2))$ .

c) funkci napravo lze napsat jako

$$2^x \cdot (\ln(x^2))^{-1} = 2^x \cdot (2 \cdot \ln(x))^{-1} = 2^{-1} \cdot 2^x \cdot (\ln(x))^{-1}$$

Tato funkce se od funkce nalevo liší jen o multiplikativní konstantu, takže obě funkce rostou asymptoticky stejně rychle. Platí tudíž všechny tři možnosti:

$$2^x \cdot (\ln(x))^{-1} \in O(2^x \cdot (\ln(x^2))^{-1})$$

$$2^x \cdot (\ln(x))^{-1} \in \Theta(2^x \cdot (\ln(x^2))^{-1})$$

$$2^x \cdot (\ln(x))^{-1} \in \Omega(2^x \cdot (\ln(x^2))^{-1}),$$

takže situace (se škrtnutým symbolem náležení) v zadání nemůže nastat, tudíž je nutno prázdné místo proškrtnout.

## 20.

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

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

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

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

Řešení:

a) funkce na pravé straně je součtem dvou funkcí, z nichž druhá --  $\ln(x)$  -- roste asymptoticky pomaleji než první --  $x^2$ . Celkem tedy funkce na pravé straně roste asymptoticky stejně rychle jako  $x^2$ . Funkce na levé straně však díky multiplikativnímu členu  $\ln(x^2)$  roste asymptoticky rychleji než  $x^2$ .

Tudíž platí právě

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

b) funkce na pravé straně obsahuje exponenciální člen, funkce na levé straně jen člen polynomiální a logaritmický, jež vesměs rostou asymptoticky pomaleji než exponenciála, takže funkce na pravé straně roste asymptoticky rychleji, platí tedy právě:

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

c) podle podobné úvahy jako v b) roste funkce na pravé straně asymptoticky rychleji než funkce na levé straně. Tudíž funkce na levé straně neroste ani asymptoticky stejně rychle ani neroste asymptoticky rychleji. Formálně vyjádřeno:

$$x^3 \cdot \ln(x^2) \notin \Theta(\ln(x^2) + 2^x),$$

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

**21.**

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

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

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

**Řešení**

První vztah říká, že  $f(x)$  musí růst asymptoticky rychleji než  $g(x)$ , třetí vztah říká, že  $f(x)$  musí růst asymptoticky rychleji než  $h(x)$ . Druhý vztah říká, že asymptotická rychlost růstu  $g(x)$  a  $h(x)$  není stejná. Můžeme tedy volit např.:

$$f(x) = x^2$$

$$g(x) = x$$

$$h(x) = \ln(x).$$

**22.**

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

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

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

**Řešení**

První vztah říká, že  $f(x)$  roste asymptoticky rychleji než  $g(x)$ , druhý vztah říká, že  $g(x)$  roste asymptoticky pomaleji než  $h(x)$ , celkem tedy  $g(x)$  roste asymptoticky pomaleji než  $f(x)$  i  $h(x)$ . Poslední vztah pak říká, že  $f(x)$  a  $h(x)$  nerostou asymptoticky stejně rychle. Můžeme tedy volit např.:

$$f(x) = x^2$$

$$g(x) = \ln(x)$$

$$h(x) = x.$$