Řešení

Tranzitivita

$$
\begin{aligned}
0 \le f(n) \le c_1 g(n) & \\
0 \le g(n) \le c_2 h(n) & \qquad \text{z definice} \\
\\
0 \le c_1 g(n) \le c_1 c_2 h(n) & \qquad \text{vynásobím druhou nerovnici } c_1 \\
0 \le f(n) \le c_1 g(n) \le c_1 c_2 h(n) & \qquad \text{spojím s první nerovnicí} \\
\\
c_3 = c_1 c_2 & \qquad \text{zadefinuji si konstantu } c_3 \\
0 \le f(n) \le c_3 h(n) & \qquad \square
\end{aligned}
$$

Exponenciely

a) Tvrzení *platí.*

Funkci 2^{n+1} můžeme rozepsat jako $2 \cdot 2^n$. Podle definice tedy musí platit:

$$\exists n_0 \in \mathrm{N}^+, \exists c > 0, \forall n \ge n_0 : \quad 0 \le 2 \cdot 2^n \le c \cdot 2^n,$$

což platí triviálně pro $c \ge 2$ a $n_0 >= 1$.

b) Tvrzení *neplatí.*

Podle definice tvrdíme, že musí platit:

$$\exists n_0 \in \mathrm{N}^+, \exists c > 0, \forall n \ge n_0 : \quad 0 \le 2^{2n} \le c \cdot 2^n.$$

Upravíme nerovnici a vyjádříme c:

$$
\begin{aligned}
2^{2n} &\le c \cdot 2^n \\
\log_2 2^{2n} &\le \log_2(c \cdot 2^n) \\
2n \log_2 2 &\le \log c + \log_2 2^n \\
2n \log_2 2 &\le \log c + n \log_2 2 \\
2n &\le \log c + n \\
n &\le \log c \\
2^n &\le c
\end{aligned}
$$

Z nerovnosti vidíme, že konstanta c by musela být vyšší než 2^n pro libovolně velké n, vztah tedy neplatí. (Např. 4^n tedy není $\mathcal{O}(2^n)$).

Jednotková matice

První blok programu se provede n^2-krát, druhý blok programu n-krát. Počet kroků můžeme pro zadané n vyjádřit vzorcem $n^2 + n$, který je zároveň horní, dolní a tedy i těsnou mezí na počet kroků. U asymptoticky těsné meze zanedbáme pomaleji rostoucí člen posloupnosti n, asymptoticky těsnou mezí počtu kroků v závislosti na n je tedy $\Theta(n^2)$.

Hvězdičky

Program vytiskne v každé iteraci cyklu přesně jednu hvězdičku. Zjišťuji tedy, kolikrát se cyklus provede v závislosti na n. Podle podmínky vidíme, že cyklus se zastaví, pokud hodnota proměnné i přesáhne n^2.

$$\begin{aligned}
i &= 2^j && \text{vyjádřím si hodnotu } i \text{ v } j\text{-té iteraci} \\
n^2 &\leq 2^j && \text{ptám se: "kolik iterací potřebuju na to, aby } i \text{ bylo} \geq n^2 \text{ a cyklus tedy skončil?"} \\
\log_2 n^2 &\leq \log_2 2^j && \text{odpověď zjistím, když si vyjádřím } j \\
2 \log_2 n &\leq j \log_2 2 \\
2 \log_2 n &\leq j
\end{aligned}$$

Pro ukončení cyklu musí být počet iterací j větší než $2 \log_2 n$, program tedy vytiskne $\lceil 2 \log_2 n \rceil$ hvězdiček.

Při určení asymptotické meze zanedbám zaokrouhlování a konstanty. Asymptotická horní mez časové složitosti programu je tedy $\mathcal{O}(\log n)$.

Mince

Nejpve se zamyslíme, jaký je nejvyšší počet mincí, kdy můžeme s rovnoramennými váhami najít lehčí minci na jedno zvážení. Jsou to *tři* mince. Jak? Zvážíme dvě z nich: buď je to mince na levé misce vah, na pravé misce vah, nebo mince, kterou jsme nezvážili.

Obdobně můžeme pokračovat i se skupinami mincí. Konkrétně pokud máme devět mincí, můžeme si je rozdělit na tři hromádky po třech a přechozí krok opakovat dvakrát – nejdřív zjistíme, ve které z hromádek se nachází lehčí mince, a pak krok opakujeme s mincemi z lehčí hromádky. Na 9 mincí tedy potřebujeme *2 vážení.*

Snadno nahlédneme, že tento postup se dá zobecnit. Pokud máme k dispozici i třetí krok, můžeme v prvním kroku rozdělit mince na třetiny, v druhém kroku rozdělit lehčí třetinu opět na třetiny, a pokud nám zbyly v lehčí třetině nanejvýš tři mince, tak pomocí posledního vážení odhalit falešnou minci. Protože $27/3 = 9$, tento postup zafunguje při $n = 27$.

Obecně je algoritmus v podstatě "ternární vyhledávání". V každém kroku dělíme mince na třetiny, a tedy pokud máme k dispozici i vážení, dokážeme odhalit falešnou minci mezi $n \leq 3^i$ mincemi. (Vyzkoušejte si, že vážení funguje i ve chvíli, kdy počet mincí není přesně roven mocnině tří.)