Cvičení 9

Příklad 1: Rozeberte a zdůvodněte, jakou časovou složitost (v asymptotické notaci) má následující problém: Daná je posloupnost z čísel $1, 2, \ldots, n$ (s možným opakováním i chybějícími čísly). Úkolem je zjistit, zda tato posloupnost je permutací.

Příklad 2: Odhadněte asymptoticky výsledek rekurence

$$T(n) = T(n/2) + T(n/3) + T(n/4) + 5n^2 \,.$$

Příklad 3: Seřaďte následující tři funkce podle asymtpotické rychlosti jejich růstu od nejpomalejšího růstu.

a) $n + \sqrt{n} \cdot \log n$

b) $n \cdot \log n$

c) $\sqrt{n} \cdot \log^2 n$

Příklad 4: Seřaďte následující tři funkce podle asymtpotické rychlosti jejich růstu od nejpomalejšího růstu.

a) 2^n

b) $2^{\sqrt{n}}$

c) $n!$

Příklad 5: Seřaďte následující tři funkce podle asymtpotické rychlosti jejich růstu od nejpomalejšího růstu.

a) $n/2005$

b) $\sqrt{n} \cdot 3n$

c) $n + n \cdot \log n$

Příklad 6: Seřaďte následující tři funkce podle asymtpotické rychlosti jejich růstu od nejpomalejšího růstu.

a) $(\log n)^n$

b) n^n

c) $2^{\sqrt{n}}$

Příklad 7: Nezáporná funkce $T(n)$ splňuje pro všechna přirozená n následující rekurentní vztah

$$T(n) \leq 3 \cdot T(n/5) + n\,.$$

Jaký je (nejlepší) asymptotický odhad růstu funkce $T(n)$ v závislosti na n?

Příklad 8: Nezáporná funkce $T(n)$ splňuje pro všechna přirozená n následující rekurentní vztah

$$T(n) \leq 4 \cdot T(n/2) + n\,.$$

Jaký je (nejlepší) asymptotický odhad růstu funkce $T(n)$ v závislosti na n?

Příklad 9: Nezáporná funkce $T(n)$ splňuje pro všechna přirozená n následující rekurentní vztah

$$T(n) \leq T(n/3) + T(n/4) + T(n/5) + n\,.$$

Jaký je (nejlepší) asymptotický odhad růstu funkce $T(n)$ v závislosti na n?

Příklad 10: Nezáporná funkce $T(n)$ splňuje pro všechna přirozená n následující rekurentní vztah

$$T(n) \leq T(n/2) + T(n/3) + T(n/6) + n\,.$$

Jaký je (nejlepší) asymptotický odhad růstu funkce $T(n)$ v závislosti na n?

Příklad 11: Uvažujme následující problém, kde vstupem je nějaká množina booleovských proměnných $V = \{x_1, x_2, x_n\}$ a výstupem booleovská formule φ (vytvořená z proměnných z množiny V a booleovských operátorů $\wedge$, $\vee$ a $\neg$) taková, že φ nabývá hodnoty TRUE právě tehdy, když právě jedna z proměnných z množiny V nabývá hodnoty TRUE.

Jedním z možných řešení je následující rekurzivní algoritmus:

- Nechť $n = |V|$. Pokud $n = 1$, vrať (jedinou) proměnnou z V.
- Pokud $n > 1$:
  - Rozděl V na množiny L a R takové, že $L \cup R = V$, $L \cap R = \emptyset$, $|L| = \lceil n/2 \rceil$ a $|R| = \lfloor n/2 \rfloor$.
  - Rekurzivně vytvoř formule φ_L a φ_R pro množiny L a R.
  - Vrať formuli

$$(\varphi_L \wedge \bigwedge_{x_i \in R} \neg x_i) \vee (\varphi_R \wedge \bigwedge_{x_i \in L} \neg x_i)$$

Co nejpřesněji odhadněte velikost výsledné formule. Pro jednoduchost počítejte, že velikost názvu proměnné je v $\Theta(1)$.

Poznámka: Zápis $\bigwedge_{x_i \in X} x_i$ označuje formuli $x_1 \wedge x_2 \wedge \cdots \wedge x_k$, kde $X = \{x_1, x_2, \ldots, x_k\}$.

Příklad 12: Jak jistě víte, vynásobit dvě n-místná čísla školním postupem (každou číslici s každou) trvá čas $\Theta(n^2)$. My si slovně popíšeme rychlejší rekurzivní algoritmus. Předpokládejme, že $n = 2k$ je velmi velké číslo (pokud je n liché, doplníme nulu). Součin $a \cdot b$ dvou n-místných čísel a, b vypočítáme následovně:

- Rozdělíme obě čísla na poloviční úseky jejich dekadických zápisů, tj. $a = 10^k \cdot a_1 + a_2$ a $b = 10^k \cdot b_1 + b_2$. Jednoduše upravíme součin $a \cdot b = (10^k \cdot a_1 + a_2)(10^k \cdot b_1 + b_2) = 10^n(a_1 \cdot b_1) + 10^k(a_1 b_2 + a_2 b_1) + (a_2 \cdot b_2)$. Takže pro výpočet $a \cdot b$ nám stačí spočítat každý ze třech výrazů v posledních závorkách.
- Rekurzivní aplikací téhož algoritmu násobení spočítáme $z_1 = (a_1 \cdot b_1)$ i $z_3 = (a_2 \cdot b_2)$.
- Dále jednoduchým sečtením a následnou rekurzivní aplikací algoritmu násobení spočítáme $(a_1 + a_2) \cdot (b_1 + b_2)$. Z toho už jednoduchým odečtením vypočítáme hodnotu poslední požadované závorky $z_2 = (a_1 b_2 + a_2 b_1) = (a_1 + a_2) \cdot (b_1 + b_2) - z_1 - z_3$.
- Mezivýsledky sčítáním složíme do výsledného $a \cdot b = 10^n z_1 + 10^k z_2 + z_3$.

Jaká je časová složitost popsaného algoritmu?

*__Příklad 13:__ Chytrý Strassenův algoritmus pro násobení dvou matic velikosti $n \times n$ pracuje zhruba následovně: Každá matice A, B se rozdělí do čtyř podmatic polovičních velikostí a součin $A \times B$ se rozepíše pomocí známých pravidel násobení a chytrého triku do sedmi součinů a několika součtů mezi zmíněnými polovičními podmaticemi. Jaká je časová složitost takového algoritmu?

*__Příklad 14:__ Jak byste v Příkladě 12 odvodili výsledný asymptotický odhad na $T(n)$ bez zanedbání "+1" v $T(k + 1)$?