Příklad (mult)
Ukažte, dva (velmi podobné) algoritmy, jak je možno násobit n ciferná čísla v čase $O(n^{log_2 3})$.

Řešení (První)
BÚNO je n mocnina dvou (pokud není, tak čísla doplníme na začátku nulami a zvětšíme tak jejich délku maximálně na $2n$). Čísla tedy rozdělíme na poloviny a zapíšeme jako $\overline{AB} = A \cdot 2^{n/2} + B$ a $\overline{CD} = C \cdot 2^{n/2} + D$. Tedy součin těchto čísel je roven $\overline{AB} \cdot \overline{CD} = A \cdot C \cdot 2^n + B \cdot D + (A \cdot D + B \cdot C) \cdot 2^{n/2}$.

Pokud bychom počítali přímo tyto součiny, tak jsme si moc nepomohli, protože pokud budeme počítat 4krát poloviční vstupy, pak podle 1. příkladu bude složitost $n^{\log_2(4)} = n^2$ (jelikož $4 \cdot (1/2)^\beta = 1 \implies \beta = \log_{1/2}(1/4) = \log_2(4)$). My si ale můžeme všimnout, že když vezmeme $(B - A) \cdot (c - D) = (A \cdot D + B \cdot C) - A \cdot D - B \cdot C$, tak stačí přičíst jen 2 hledané součiny a máme přesně ten součet třetího a čtvrtého, který chceme. Zároveň rozdíl jistě zachová velikost čísla, tedy i tento 'větší' součin je součin stále polovičních (co do počtu cifer) čísel.

Tedy náš algoritmus vrátí součin daných čísel pokud je $n = 1$, jinak spustí sám sebe na čísla A, C, čísla B, D a čísla $B - A, C - D$. Až dostane výsledky $X = A \cdot C$, $Y = B \cdot D$ a $Z = (B - A) \cdot (c - D)$, tak spočítá $X \cdot 2^n + Y + (Z + X + Y) \cdot 2^{n/2}$.

Násobením 2 $n/2$ciferných čísel můžeme dostat až $n$ciferné číslo, tedy X, Y, Z jsou '$n$ciferná'. Navíc násobením 2^n se můžeme dostat až na $2n$ciferné číslo. Sčítání a násobení mocninou 2 (tedy bitový posun) zřejmě probíhá v lineárním čase k velikosti vstupu (tj. v $O(2n) = O(n)$), jelikož prochází pole cifer jenom jednou. Tedy v jednom kroku děláme $O(n)$ operací a voláme se na 3 poloviční vstupy, tedy rekurence $t(n) \leq O(n) + 3 \cdot t(n/2)$, tj. náš algoritmus běží v čase $O(n^{\log_2(3)})$, jelikož $3 \cdot (1/2)^\beta = 1 \implies \beta = \log_2(3)$.

Řešení (Druhý)
V podstatě úplně stejný algoritmus funguje pomocí součtů místo rozdílu, jelikož $(A + B) \cdot (C + D) - A \cdot C - B \cdot D = A \cdot D + B \cdot C$. Tento nepatrný rozdíl přináší však jednu nepříjemnost, a to, že součet může mít i o bit více než původní čísla. To však můžeme vyřešit tak, že si zapamatujeme dvě hodnoty $F, G \in \{0, 1\}$ podle toho, zda přeteklo sčítání $(A + B)$ resp. $(C + D)$ (součty však spočítáme jen s $n/2$ bitovou přesností). Výsledek pak vždy spočítáme jako $X \cdot 2^n + Y + (Z - X - Y) \cdot 2^{n/2} + (C + D) \cdot F \cdot 2^n + (A + B) \cdot E \cdot 2^n$. Složitost tak zůstane $O(n^{\log_2(3)})$.