

1. $\mathcal{O}(\log n) = \mathcal{O}(\log n^2) = \mathcal{O}(\log_{\pi} n) \subset \mathcal{O}(\log^2 n) \subset \mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n) = \mathcal{O}(2n) \subset \mathcal{O}(n^2) \subset \mathcal{O}(2^n) = \mathcal{O}(2^{n+1}) \subset \mathcal{O}(2^{n+\log n}) \subset \mathcal{O}(n!) \subset \mathcal{O}(n^n) \subset \mathcal{O}(n^{n+1}) \subset \mathcal{O}(n^{2n})$
2. Z definice máme c_1 a n_1 t.ž. $\forall n \geq n_1: f(n) \leq c_1 g(n)$ a c_2 a n_2 t.ž. $\forall n \geq n_2: g(n) \leq c_2 h(n)$. Z tranzitivity $\leq$ vidíme, že platí $f \leq c_1 g(n) \leq c_1 c_2 h(n)$ ($\forall n \geq n_1 \wedge n \geq n_2$).
Nastavením $c = c_1 c_2$ a $n_0 = \max\{n_1, n_2\}$ dostáváme $f \in \mathcal{O}(h)$ z definice.
3. Nejdřív dokážeme pomocné tvrzení: $\mathcal{O}(f) \in \mathcal{O}(f') \wedge \mathcal{O}(g) \in \mathcal{O}(g') \Rightarrow \mathcal{O}(f + g) \in \mathcal{O}(f' + g')$:
Máme c_f, c_g t.ž. pro každé n dostatečně velké platí $f(n) \leq c_f f'(n)$ a $g(n) \leq c_g g'(n)$. Sečtením dostaneme $f(n) + g(n) \leq c_f f'(n) + c_g g'(n)$. Pro $c = \max(c_f, c_g)$ pak platí $(f + g)(n) \leq c(f' + g')(n)$ (pravou stranu jsme pouze zvětšili).
Zjevně $f \in \mathcal{O}(\max(f, g))$ a $g \in \mathcal{O}(\max(f, g))$. Pak z pomocného tvrzení platí $f + g \in \mathcal{O}(2 \max(f, g)) = \mathcal{O}(\max(f, g))$.
4. Ne, protipříklad:
 $f(n) = 0$ pro $2 \mid n$ a $f(n) = 1$ pro $2 \nmid n$
 $g(n) = 1$ pro $2 \mid n$ a $g(n) = 0$ pro $2 \nmid n$.
Pro libovolné c a n_0 můžeme vzít nějaké $n \geq n_0$ liché jako protipříklad pro $\forall n \geq n_0: f(n) \leq c g(n)$ a nějaké sudé pro opačný případ.
5. Vyberme libovolný vrchol grafu u a najdeme od něj nejvzdálenější vrchol v . Pak od tohoto vrcholu opět najdeme nejvzdálenější vrchol w , čímž dostáváme nejdelší cestu ve stromě vw (z každého vrcholu u je nejvzdálenější vrchol v jeden z konců nejdelší cesty, jinak by bylo možné nejdelší cestu prodloužit o cestu do v a pak do w). Kořen stromu musí ležet v půlce této cesty.

6. Cílem je najít takovou posloupnost, aby pivot byl co nejbližší kraji (pod)posloupnosti, čímž dosáhneme n fází (a $O(n^2)$ celkové složitosti). Lze například použít permutaci $\{1, \dots, n\}$, kde prostřední prvek je vždy ten nejmenší.

Podoba posloupnosti závisí na konkrétní implementaci QuickSortu, ale pokud předpokládáme, že „prostřední“ prvek je $\lfloor \frac{n}{2} \rfloor$ -tý a že prvky větší než pivot vybíráme stabilně, tak pro $n = 8$ může vypadat takto: 6, 4, 2, 1, 3, 5, 7, 8.