1. *Nekuchaři:*

   - $T(n) = 2T(n/2) + \Theta(n \log n)$ *a* $T(1) = 1$,
   - $T(n) = n^{1/2} \cdot T(n^{1/2}) + \Theta(n)$ *a* $T(2) = 1$.

   Oba příklady vyřešíme pomocí rozboru stromu rekurzivních volání.

   V prvním případě je na i-té hladině 2^i podproblémů velikosti $n/2^i$ a celkový čas na hladině tedy bude $2^i \cdot n/2^i \log(n/2^i) = n \log(n/2^i)$. Tyto časy tedy sečteme přes všechny hladiny

   $$T(n) = \sum_{i=0}^{\log n} n \log(n/2^i) = n \sum_{i=0}^{\log n} \log(n) - i = n \cdot (1 + 2 + \cdots + \log n) = \Theta(n \log^2 n).$$

   Pro druhou rekurenci je na i-té hladině $n^{1-1/2^i}$ podproblémů velikosti $n^{1/2^i}$ (ověřte si dosazením). Celkový čas na hladině tedy vyjde přesně n a jediná otázka je kolik budeme potřebovat hladin než dostaneme problémy konstantní velikosti, neboli pro jaké k je $n^{1/2^k} \leq 2^*$.

   $$n^{1/2^k} = 2 \iff \log n \cdot 1/2^k = 2 \iff \log\log n - k = 1 \iff k = \log\log n - 1$$

   A tedy řešením je $T(n) = \Theta(n \log\log n)$.

2. *Skoroskoromediány: Student Šťoura si místo skoromediánů za pivoty volí „skoroskoromediány", které leží v prostředních šesti osminách vstupu. Jaké dosahuje časové složitosti?*

   Pro Quickselect nám stačí, že velikost podproblému na i-té hladině je nejvýše $(7/8)^i n$. Tedy celková složitost je méně než $n + 7/8n + (7/8)^2 n + (7/8)^3 n + \cdots = O(n)$.

   Pro Quicksort je klíčové pozorování, že nezávisle na volbě pivota je součet velikostí všech podproblémů na i-té hladině nejvýše n - každý prvek je vždy přiřazen do nejvýše jedné z větví. Tedy čas strávený na jedné hladině bude vždy $O(n)$ a stačí nám spočítat počet hladin. Ptáme se pro jaké k je $(7/8)^k n \leq 1$, což nastane pro $k = \log_{8/7} n = O(\log n)$. A tedy i se skoroskoromediány třídíme v čase $O(n \log n)$.

3. *Průměrný pivot:* Jak by dopadlo, kdybychom na vstupu dostali posloupnost reálných čísel a jako pivota používali aritmetický průměr?

   Obecně si tím nemusíme vůbec pomoci. Pro libovolnou posloupnosti $a_1, a_2, \ldots a_{n-1}$ umíme najít číslo a_n tak velké, že průměr $a_1, a_2, \ldots, a_n$ bude větší než všechna a_i pro $i \leq n-1$. Pokud začneme s $a_1 = 1$, můžeme takto induktivně sestrojit posloupnost délky n, kde aritmetický průměr vždy oddělí pouze největší prvek od zbytku. Tedy pokud bychom za pivota volili aritmetický průměr, budeme potřebovat n hladin rekurzivních volání u Quickselectu i Quicksortu.

4. *Linearselect:* Proč při vybírání k-tého nejmenšího prvku používáme zrovna pětice? Fungoval by algoritmus s trojicemi? Nebo se sedmicemi? Byl by pak stále lineární?

   Aplikujeme stejný argument jako v Průvodci, jen vyměníme pětice za trojice resp. sedmice. Pro trojice dostáváme, že pro pivota p zvoleného jako medián z mediánů trojic platí že alespoň $n/3$ prvků je menších a alespoň $n/3$ prvků je větších. Tím pádem sestrojíme rekurenci

   $$T(n) = T(n/3) + T(2n/3) + O(n).$$

*Nemá smysl se ptát pro jaké k je $n^{1/2^k} = 1$, jelikož to nenastane nikdy.

Tato rekurence ale bohužel vede na časovou složitost $O(n \log n)$.

Naopak pokud bychom pivota zvolili jako medián z mediánů sedmic pak alespoň $2/7n$ prvků je menších než p a alespoň $2/7n$ prvků je větších než p. Tím dostaneme rekurenci

$$T(n) = T(n/7) + T(5/7n) + O(n)$$

Tuto rekurenci můžeme stejně jako pro pětice vyřešit uhádnutím $T(n) = cn$ a zjištěním že rovnost platí pro $c = 7$.

Zbylé příklady si schováváme na příště.