1. *QuickSelect.* Co kdybychom v QuickSelectu používali jako pivota aritmetický průměr čísel na vstupu? Jaká bude časová složitost v nejhorším a nejlepším případě?

A jak by to dopadlo, kdybychom vždy vybírali za pivot „skoroskoromedián“, což je prvek, který leží v prostředních šesti osminách vstupu?

2. *Pomocná paměť pro QuickSort.* Jak velkou potřebuje QuickSort pomocnou paměť v nejlepším a nejhorším případě? Předpokládejme, že třídíme na místě, tedy každé rekurzivní volání dostane jen indexy $i < j$ v poli a má setřídit úsek od i do j.

Navrhněme následující úpravu: Místo rekurze použijeme vlastní zásobník na úseky, které se mají setřídit. Navíc při rozdělení pole pivotem na dva úseky dáme jen ten *delší* úsek na zásobník a kratší úsek rovnou setřídíme. Dokažte, že na zásobníku máme vždy jen $O(\log n)$ úseků.

3. *Nejlevnější vrcholové pokrytí stromu.* Dáno městečko ve tvaru stromu (ulice odpovídají hranám a křižovatky vrcholům). Chceme na křižovatky umístit strážníky, aby každá ulice měla strážníka alespoň na jednom konci. Každá křižovatka má danou cenu, za kterou na ní lze umístit strážníka. Cílem je najít nejlevnější rozmístění strážníků.

(Výběru podmnožiny vrcholů grafu takové, že každá hrana má alespoň jeden konec vybraný, se říká vrcholové pokrytí.)

4. *Triangulace mnohoúhelníku.* Konvexní mnohoúhelník můžeme triangulovat, tedy rozřezat neprotínajícími se úhlopříčkami na trojúhelníky. Nalezněte takovou triangulaci, aby součet délek řezů byl nejmenší možný

5. *Medián dvou setříděných polí.* Mějme dvě setříděné posloupnosti (ne nutně stejně dlouhé), reprezentované v poli. Jak najít jejich medián v sublineárním čase (tedy $o(n)$)?

6. Dána matice $n \times n$ celých čísel A, která je v každém řádku i každém sloupci rostoucí. Zjistěte, jestli existují indexy i, j takové, že $A_{i,j} = i + j$.