

1. *(Ne)stabilita*: Jak zajistit pro nestabilní algoritmus, aby třídil stabilně?
2. *Knížky v knihovně*: Mějme posloupnost, která vznikla ze setříděné tím, že jsme přesunuli k prvků. Navrhněte algoritmus, který ji co nejrychleji dotřídí. Pozor na to, že k předem neznáme. Můžete nicméně předpokládat, že k je mnohem menší než délka posloupnosti.
3. *Opakované prvky*: Navrhněte algoritmus na zjištění, jestli se v zadané n -prvkové posloupnosti opakují některé prvky. Dokažte problém vyžaduje v porovnávacím modelu čas alespoň $\Omega(n \log n)$.
4. *Součet*: Mějme množinu čísel a číslo x . Zjistěte, zda množina obsahuje dvojici prvků se součtem x .
5. *Konstantní skoky*: Uvažujme hešování řízené obecnější lineární posloupností $h(x, i) = (f(x) + c \cdot i) \bmod m$, kde c je konstanta nesoudělná s m . Srovnajte jeho chování s obyčejným lineárním přidáváním.
6. *Delete lépe*: Jak byste pro hashování s otevřenou adresací a lineárním přidáváním naimplementovali delete, pokud chcete, aby po nalezení políčka, ze kterého se má mazat, už trval amortizovaně $O(1)$?