

1. Mějme množinu přirozených čísel A a číslo x . Jak zjistit, zda A obsahuje dvojici prvků se součtem x ?
 2. Mějme hešovací funkci $h : [U] \rightarrow [m]$. Pokud o této funkci nic dalšího nevíte, kolik pokusů potřebujete, abyste našli k -tici prvků, které se všechny zobrazí do téže přihrádky?
 3. Pokud bychom chtěli hešovat řetězce 8-bitových znaků, můžeme použít některou z hešovacích funkcí pro posloupnosti. Jaké problémy mohou vyvstat, pokud všechny řetězce nejsou stejně dlouhé a jak je vyřešit?
 4. 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.
-
5. Mějme rovnoramenné váhy a 12 kuliček, z nichž právě jedna je těžší než ostatní. Na miskou lze dát i více kuliček naráz. Navrhněte, jak na 3 porovnání najít těžší kuličku.
 6. Mějme rovnoramenné váhy a 12 kuliček, z nichž právě jedna je lehčí nebo těžší než ostatní. Na miskou lze dát i více kuliček naráz. Navrhněte, jak na 3 porovnání najít tuto jinou kuličku.
 7. Co když kuliček v předchozím příkladu bude 13?

Hešovací funkce Hešování řetězců můžeme zrychlit tím, že čtveřice znaků prohlásíme za 32-bitová čísla a zahešujeme posloupnost čtvrtinové délky. Naprogramujte takovou hešovací funkci a nezapomeňte, že délka řetězce nemusí být dělitelná čtyřmi. **(10 bodů)**

Mergesort Zkuste si napsat mergesort bez rekurze ve Vašem oblíbeném programovacím jazyce. **(14 bodů)**