(a, b)-strom: Vícecestný vyhledávací strom s parametry $a \geq 2, b \geq 2a - 1$, kde kořen má 2 až b synů a každý jiný vrchol má a až b synů (tedy $a - 1$ až $b - 1$ klíčů).

Hashovací tabulka s přihrádkami: Datová struktura na udržování množiny prvků, která prvek uloží do přihrádky (typicky spojového seznamu) podle jeho „hashe“ – malého výsledku deterministické hashovací funkce.

1. Ukažte, proč jsou nutné podmínky na parametry a, b u (a, b)-stromu – vytvořte příklady, kde některá z operací nepůjde provést nebo bude pomalá.
2. Popište asymptotickou složitost operací na (a, b)-stromě v závislosti na počtu uložených prvků n a parametrech a, b.
3. Najděte optimální parametry pro (a, b)-strom 64bitových čísel ukládaný na disk s 4096bytovými bloky (tedy každé čtení/zápis načte/zapíše 4096 bytů, chceme minimalizovat jejich počet).
4. Upravte `Insert` na $(2, 4)$-stromě tak, aby mu stačil pouze jeden průchod stromem od kořene dolů.[1]
5. Mějme softwarový systém, který používá hashovací tabulku s přihrádkami s fixní hashovací funkcí h pro řetězce bytů pevné délky l definovanou polynomem ($h(x) = \sum_{i=1}^{l} a^i x_i \bmod m$). Představte si, že jste útočník, který chce systém zahltit. Najděte pro to vhodnou posloupnost prvků. Jaká bude asymptotická složitost `Insert`u každého prvku v takovém případě, jaká pak následného `Find`u?
6. Rozšiřte nějaký vyhledávací strom přiřazující klíčům čísla tak, aby uměl operaci `Add`(x, a, b), která ke každému číslu s klíčem v intervalu $[a, b]$ přičte hodnotu x, v čase $O(\log n)$. Zachovejte složitost existujících operací.

[1] Můžete si rozmyslet takovou úpravu i pro `Delete`. Tím se $(2, 4)$-strom „vyrovná“ obvyklé implementaci vesměs ekvivalentního červeno-černého stromu.

