Cviko z Počítačové algebry

2 Měření času v Sage, základní algoritmy

1. Pomocí `timeit` změřte rychlost sečtení $1+1$ jakožto
   (a) prvků okruhu $\mathbb{Z}$ (`ZZ`),
   (b) nativních pythonových integerů,
   (c) nativních pythonových floatů,
   (d) prvků okruhu $\mathbb{Z}_2$ (`Zmod(2)`).

   Liší se výsledky, pokud se sčítají přímo „napsaná čísla", oproti tomu, když se sčítají proměnné, do kterých se čísla uložila?

2. Napište dvě funkce, které pro $a, b \in \mathbb{Z}$ spočítají $\mathrm{NSD}(a, b)$ pomocí Eukleidova algoritmu – jedna s pomocí rekurze, druhá bez ní. Ujistěte se, že tyto funkce fungují pro
   - celá čísla (i záporná),
   - polynomy – nejprve vytvořte obor polynomů nad $\mathbb{Q}$ pomocí `R.<x> = QQ[]` (pozor, tím dostane `x` speciální význam) a pak můžete zadávat polynomy jako `p = 3*x^2-2/3` atd. Zkuste nějaké *soudělné* polynomy. Proč jsou výsledky tak škaredé?

   Dále porovnejte jejich rychlost navzájem a oproti zabudované funkci `gcd`, přičemž
   (a) zadejte pevné číselné vstupy,
   (b) generujte náhodné číselné vstupy (`ZZ.random_element`) až do 2^{100}.

3. Reprezentujme číslo $n = \sum_{i=0}^{k} a_i B^i$ v bázi B pomocí seznamu (list) $[a_0, a_1, \ldots, a_k]$. Pro takto reprezentovaná čísla napište funkce sčítání a odčítání. Jedna z možností je např. stylem `nazev_fce(a,b,B)`, kdy funkci předáme dva listy na sečtení a specifikujeme bázi B. Pozor na to, že odečítáním může vyjít záporné číslo – to je potřeba nějak ošetřit.

4. Pro reprezentaci čísel z úlohy 3 se ujistěte, že umíte převádět mezi bázemi – prvky `ZZ` mají metodu `digits`, opačným směrem funguje `ZZ(list,B)`. (Není smyslem tohoto bodu programovat onen převod „od nuly".)

5. Pro reprezentaci čísel z úlohy 3 napište funkci násobení pomocí klasické školské metody. Nejprve bude potřeba napsat funkci násobení „cifrou" (konstantou).

6. Pro reprezentaci čísel z úlohy 3 implementujte metodu násobení pomocí Karacubova triku. Porovnejte rychlost s klasickým školním násobením.