



Programování 1 (NPRG030)

Cvičení 8: Rekurzivní funkce

Cvičení 8.3: Binární vyhledávání rekurzivně

- ❖ Vytvořte *rekurzivní* verzi funkce `binary_search` pro *binární vyhledávání* prvku v setříděném poli
- ❖ Vstupem funkce bude *seznam celých čísel* (tj. pole)
- ❖ Výstupem bude *index hledaného prvku*

Cvičení 8.4: Rekurzivní generování

- ❖ Napište funkci v jazyce Python, která pro *zadané k a n vrátí všechny variace k -té třídy z prvků množiny $\{1, 2, \dots, n\}$ bez opakování*

Cvičení 8.5: Rekurzivní generování II

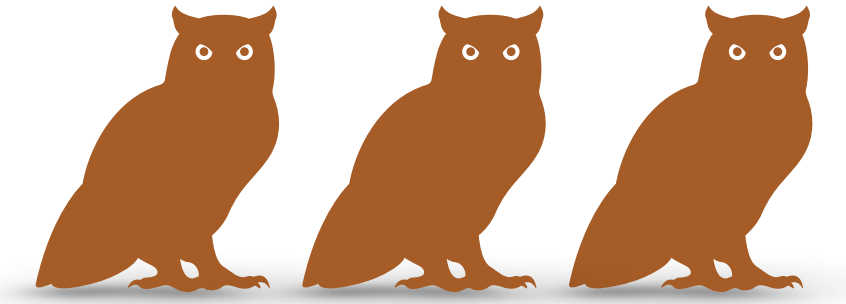
- ❖ Na vstupu je *posloupnost n přirozených čísel a* přirozené číslo *výsledek*
- ❖ Mezi čísla (operandsy) *doplňte operátory $\{ + , - \}$* tak, aby hodnota takto vytvořeného výrazu byla rovna zadanému výsledku
- ❖ Navrhněte *algoritmus, který nalezne všechna řešení*
 - ❖ Můžete předpokládat, že výraz neobsahuje závorky, tj. vyhodnocuje se postupně zleva doprava

❖ **Těžší varianta**

- ❖ Chceme vygenerovat všechny plně uzávorkované výrazy, které se vyhodnotí na zadaný výsledek



Jak psát kód: Vyrovnávací paměť



- ❖ Použijte *mezipaměť* (angl. cache) pro *nákladné operace*
- ❖ Výsledky nákladných operací ukládejte do vyrovnávací paměti, abyste zvýšili výkon programu

```
1 import functools
2
3 @functools.lru_cache(maxsize=None)
4 def fibonacci(number):
5     if number <= 1:
6         return number
7     return fibonacci(number - 1) + fibonacci(number - 2)
```

Reference

- ❖ <https://naucse.python.cz/lessons/beginners/recursion/>