

# Cvičení z Algoritmizace a Programování 1

5. cvičení

Algorithmizace

# Binární vyhledávání

seznam

```
def binSearch(a, x):  
    dolni, horni = -1, len(a)  
    stred = (dolni + horni) // 2  
    while horni - dolni > 1:  
        if a[stred] == x:  
            return stred  
        elif a[stred] < x:  
            dolni = stred  
        else:  
            horni = stred  
            stred = (dolni + horni) // 2  
    return False
```

Invariant cyklu:  $a[i] == x \Rightarrow dolni < i < horni$

- Jak zjistit počet výskytů?
- Jak najít index prvního/posledního?

# Kuličky

- Máme 3 kuličky, žádné dvě nejsou stejně těžké.
- Máte k dispozici máte rovnoramenné váhy.
- Pomocí co nejmenšího počtu vážení:
  - a) určete, která z nich je nejlehčí,
  - b) určete, která z nich je „prostřední“.



# Domácí úkol – Prostřední z pěti

- Viz ReCodEx – Algoritmizace <https://recodex.mff.cuni.cz/app/assignment/0e978f20-6e0d-4432-a68b-733224956357>
- Co se po Vás chce:
  - Popište algoritmus, který najde prostřední z pěti na co nejmenší počet vážení (v nejhorším případě).
  - Hodnotí se potřebný počet vážení a srozumitelnost popisu algoritmu (a samozřejmě zda algoritmus vůbec funguje)
- Náповěda: zkuste nejdříve vyřešit úlohu „Najděte druhý nejlehčí ze čtyř“ – na 4 vážení

# Domácí úkol – Prostřední z pěti

- Upozornění:
  - Úkolem je najít algoritmus, který bude fungovat ve všech případech!
- Proto „algoritmus“ ...
  1. Kuličky označím A, B, C, D, E
  2. Porovnáám A a B; B a C; C a D; D a E
  3. Například zjistím, že  $A < B < C < D < E$
  4. Tedy prostřední je C, potřeboval jsem k tomu 4 vážení.
- ... není algoritmus a není řešením domácího úkolu!
  - Například můžete zjistit, že  $A < B < C$  a  $C > D > E$
  - Algoritmus musí fungovat ve všech případech

Programování

# Funkce

- Úlohy Stromečky X v ReCodexu