

## Efektívne algoritmy a dátové štruktúry, 1-AIN-105

### Vyučujú:

Tomáš Vinař, M-163, tomas.vinar@fmph.uniba.sk

Andrej Baláž, M-25, andrej.balaz@fmph.uniba.sk

Filip Kerák, M-25, kerak1@uniba.sk

Sabína Samporová, T-2, sabina.samporova@trojsten.sk

Marcel Palaj, T-2, marcel.palaj@trojsten.sk

**Web:** <http://compbio.fmph.uniba.sk/vyuka/eads>

**Google Classroom:** <https://classroom.google.com> kód **sgwajmw**

### Literatúra:

Brassard, Bratley: Fundamentals of Algorithmics, Prentice Hall 1995

Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms, MIT Press 2009

## Oznamy a diskusia na Google Classroom

<https://classroom.google.com> kód **sgwajmw**

- **Vaša povinnosť sledovať oznamy!!!**
- Cieľ: odpoveď na vaše otázky v najkratšom možnom čase
- Odpovedajú učitelia, študenti
- Všetky otázky sú verejné  $\Rightarrow$  žiadne detaily vlastných riešení (resp. až po termíne odovzdania danej úlohy)
- Link aj na web stránke predmetu

## O čom je tento predmet?

- Pre daný problém, úlohou je nájsť **efektívny algoritmus**, ktorý tento problém rieši.
- Rozpoznať, kedy efektívny algoritmus **neexistuje**.
- Urobiť z vás **lepších programátorov**.

## Osnova predmetu:

- Úvod, výpočtová zložitosť
- Základné dátové štruktúry
- Techniky tvorby efektívnych algoritmov  
(greedy algoritmy, dynamické programovanie, rozdeľuj a panuj)
- NP-ťažké problémy

## Hodnotenie predmetu

- 30%: Domáce úlohy (cca každé dva týždne)  
(vr. dvoch programátorských príkladov)
- 20%: Midterm (21. novembra 2023, 18:10-20:10)
- 50%: Písomná skúška
- zo skúšky je potrebné získať aspoň 50% bodov
- 90+ = A, 80+ = B, 70+ = C, 60+ = D, 50+ = E

## Opisovanie

- Budeme kruto trestať:
  - –100% príslušného bodového hodnotenia
  - disciplinárna komisia
- **Podporujeme** diskusiu o domácich úlohách, **ale**:
  - Nerobte si poznámky
  - Počkajte niekoľko hodín, kým začnete spisovať vlastné riešenie

## Programátorské domáce úlohy

- <https://testovac.ksp.sk/>
- automatizované testovanie, neodovzdávate popis
- časová a pamäťová efektívnosť programu → časový a pamäťový limit
  - časový limit zvyčajne  $1,5 - 2 \times$  vzorového riešenia
  - nastavované pre Python a C++, ostatné jazyky na vlastné riziko
  - ak vám riešenie neprechádza, (zvyčajne) treba inú myšlienku
  - obmedzenia: (napr.  $1 \leq n \leq 100\,000$ )
- dovolené všetky prvky jazyka (knihnice, funkcie, ...), ktoré prejdú testovačom
- vstup a výstup: štandardný vstup a výstup. Zhoda znak po znaku.

## Príklady riešení

Správne:

```
a = int(input())  
b = int(input())  
print(c)
```

Nesprávne:

```
a = int(input("Zadaj cislo"))  
b = int(input("Zadaj cislo"))  
print("Sucet je ", c)
```

## Odpovede testovača

1. our 0001: 7\_10\_5\_10  
your 0001: 4\_10\_2\_10
2. our 0001: 7\_10\_5\_10  
your 0001: 7\_10\_5\_10\_
3. your terminated, our continues:  
our 0001: 0\_0\_0\_0
4. our terminated, your continues:  
your 0002: 0\_0\_0\_0

## Načítavanie vstupu zo súboru

Ak ste niekedy robili s príkazovým riadkom pomerne priamočiare:

```
python du1-3.py < ./test/1.a.in
```

Vyhnete sa problémom s načítavaním vstupu.

## Vypisovanie výstupu do súboru

Ak ste niekedy robili s príkazovým riadkom tiež pomerne priamočiare:

```
python du1-3.py < ./test/1.a.in > ./test/1.a.out
```

Takto máte súbor s výstupom a viete napríklad porovnávať dve riešenia.

## Viac info

- <https://testovac.ksp.sk/wiki/odovzdavanie-info/>
- <https://testovac.ksp.sk/wiki/odpovede-testovaca/>
- <https://testovac.ksp.sk/tasks/ls-uvod-sucetmensich/>