

# IB002 - 2. sada domácich úloh

Úloha je určená pre študentov sem. skupín IB002/05, IB002/06. Každý, kto má túto skupinu riadne zapísanú, je povinný odovzdať vypracované obidva príklady. Študenti, ktorí nepatria do týchto skupín môžu úlohu odovzdať a majú nárok na opravu, nie však na bonusové body.

**Termín odovzdania:** 23.4. do 23:42

**Zpôsob odovzdania:** elektronicky do odovzdávare v ISe (pdf dokument formátu A4) alebo papierovo na cvičeniach F. Štefaňáka. Nezabudnite i v elektronickej verzii uviesť meno a UČO.

**Vypracovanie ako dvojica:** Úlohy je možné vypracovať ako dvojica. Dvojica odovzdáva len jedno riešenie podpísané oboma menami. Body dostanú obaja z dvojice. Každý príklad je hodnotený samostatne, je teda možné vytvoriť rozdielne páry pre prvý a druhý príklad.

**Hodnotenie:** Za výborne vypracované riešenia môžete získať až dva body za príklad. Za neodovzdaný príklad, za riešenie, ktoré je extrémne nedostatočné (prázdny papier) a za opisovanie naopak 2 body za príklad strácate.

## Príklad 1

□ Vyhladať 2 radiace algoritmy, ktoré sa buď nepreberali na cvičeniach, alebo ide o nejakú ich netriviálnu obmenu. Pre každý z týchto algoritmov popíšte nasledujúce body:

- princíp algoritmu, podobnosť s preberanými algoritmami.
- prečo sa vám páči a vybrali ste si ho
- výhody (v porovnaní s druhým algoritmom), vhodné použitie
- nevýhody, nevhodné použitie

Uvedte zdroj algoritmu (url, literatúra). Neuvádzajte žiadny kód, použite len prirodzený jazyk.

Pokiaľ chcete za úlohu dostať k základu 0 bodov aj bonusové body navyše, vypracujte riešenie vo forme eseje (rozsah približne 350 slov na každý z algoritmov).

Je vhodné, aby každý z algoritmov vypracoval iný z dvojice. Pokiaľ budete riešenie vypracovávať samostatne, stačí, ak namiesto toho popíšete 1 algoritmus v rozsahu asi 500 slov a porovnanie rozvediete vzhľadom k všetkým ostatným preberaným algoritmom.

Algoritmy pre inšpiráciu: Shake sort, Shell sort, Bucket sort, Odd-even sort, Radix sort, Counting sort, Comb sort. . .

## Príklad 2

□ Máme k dispozícii štruktúru, ktorá reprezentuje strom s voľnou aritou (každý uzol má konečný počet potomkov, listy práve 0). Jednotlivé uzly sú realizované ako pár hodnoty `value` typu číslo a jednoduchého spojovaného zoznamu odkazov na uzly potomkov `firstChild`.

Navrhnite algoritmus, ktorý spočíta aritmetický priemer hodnôt všetkých uzlov v strome určenom koreňom zo vstupu. Zapište algoritmus vo vhodnom programovacom jazyku alebo pseudojazyku s jednoznačnou sémantikou. Zdôvodnite jeho korektnosť a zložitosť.

Pokiaľ chcete za úlohu dostať k základu 0 bodov aj bonusové body navyše, implementácia nesmie využívať rekurzívne volanie funkcií. Môžete použiť ľubovoľné pomocné dátové štruktúry z tých, ktoré sa v kurze preberali.