Domáca úloha č. 1

1-AIN-105, Zima 2025

Termín: 20.10.2025, 22:00

Skôr ako sa pustíte do riešenia, prečítajte si všeobecné pokyny na konci zadania.

1. [*20 bodov*] **Asymptotická zložitosť.**

a) Pre každý pár funkcií $f(n)$ a $g(n)$ v nasledujúcej tabuľke uveďte, v akom vzťahu sú tieto funkcie (vyplňte v každom políčku tabuľky áno alebo nie). V krátkosti zdôvodnite každú (aj zápornú) odpoveď.

$f(n)$	$g(n)$	$f(n) \in O(g(n))$	$f(n) \in \Theta(g(n))$	$f(n) \in \Omega(g(n))$
$n^{1.154}$	$n.\log^{154} n$			
$n\lvert\sin \pi n/2\rvert$	$\sqrt{n}$			
$n^3 + n^{3/2} + n\log n$	$\binom{n}{3}$			
$9^{\log_2 n}$	n^3			

b) Pri písaní písomky ste narazili na podobný príklad ako v časti a), akurát funkcie $f(n)$ a $g(n)$ sú zložitosti algoritmov, ktoré si žiaľ nepamätáte. Za každú správnu odpoveď je 1 bod, za nesprávnu odpoveď je -1 bod.

Aký je očakávaný počet získaných bodov, ak sa rozhodnete vypísať do tabuľky odpovede áno alebo nie náhodne?

c) Ak v časti b) navyše predpokladáte, že $f(n) = n^k$ a $g(n) = n^\ell$ (pre nejaké konštanty k a ℓ), viete navrhnúť aj lepšiu stratégiu, ktorá bude mať vyšší očakávaný počet získaných bodov? Môžete predpokladať, že prípady $k < \ell$, $k > \ell$ alebo $k = \ell$ sú rovnako pravdepodobné.

2. [*20 bodov*] **Zábava so šrúbami.** Máme krabicu A, ktorá obsahuje n rôznych šrúb a krabicu B, ktorá obsahuje n rôznych matíc. Našou úlohou je nájsť najväčšiu šrúbu a k nej prislúchajúcu maticu rovnakej veľkosti a urobiť to čo najrýchlejšie. Môžete predpokladať, že pre najväčšiu šrúbu existuje matica tej istej veľkosti a zároveň nexistuje od nej väčšia matica.

Jediná povolená operácia je vyskúšať, či konkrétna šrúba funguje s konkrétnou maticou. Výsledok takéhoto porovnania môže byť, že *sedí* (t.j. majú rovnakú veľkosť), alebo že *matica je príliš malá*, alebo že *matica je príliš veľká*. Nemáme povolené porovnávať šrúby navzájom, ani matice navzájom, lebo to môže byť nepresné.

Navrhnite algoritmus, ktorý nájde najväčšiu šrúbu a k nej príslušnú maticu, pričom použije asymptoticky čo najmenší počet skúšaní (v prípade, ak váš algoritmus používa náhodný výber, tak očakávaný počet skúšaní).

Hint 1: Slovo šrúba pochádza zo šarišského nárečia, znamená to isté ako šrób (Horná Nitra, Trenčín), šrúbek alebo šrúfek (Záhorie), šrubka (Spiš), gvužč (goralsky), гвинт (ukrajinsky), винт (rusky), či skrutka (spisovne).

Hint 2: Pri zápise vášho algoritmu môžete krabicu A aj krabicu B reprezentovať ako pole čísel—veľkostí šrúb resp. matíc. Vo vašom algoritme však môžete porovnávať len prvok poľa A s prvkom poľa B, nikdy nie dva prvky poľa A navzájom alebo dva prvky poľa B navzájom. Naopak, môžete vymieňať len prvky poľa A medzi sebou a prvky poľa B medzi sebou, nikdy nie prvky poľa A s prvkami poľa B. Výsledkom vášho algoritmu môžu byť dva indexy do polí A, B, tam kde sa nachádza maximálny prvok.

3. [*20 bodov*] **Politické kauzy (programátorská úloha).** Politici počas svojho vládnutia vyťahujú na verejnosť rôzne kauzy, aby zakryli svoje machinácie. Konkrétne, o každej kauze i vieme jej skrátený názov m_i, od ktorého dňa a_i do ktorého dňa b_i bude kauza aktuálna a jej výbušnosť v_i. (Pre jednoduchosť môžete predpokladať, že každá kauza má inú výbušnosť.) Novinári každý deň píšu výlučne len o najvýbušnejšej aktuálnej kauze. O čom budú novinári ktorý deň písať?

Vstup. Na prvom riadku vstupu sa nachádza číslo n - počet káuz. Platí, že $1 \leq n \leq 100\,000$. Nasleduje n riadkov. Každý riadok je tvorený štyroma medzerou oddelenými údajmi: `m_i a_i b_i v_i`. Platí, že `m_i` je neprázdny reťazec, ktorý sa skladá z najviac 50 malých a veľkých písmen anglickej abecedy a *ne*obsahuje medzeru. Tiež platí, že $0 \leq a \leq b \leq 100\,000$ a $1 \leq v \leq 10\,000\,000$.

Výstup. Na výstup vypíšte niekoľko riadkov formátu `nazov a b` - najvýbušnejšie kauzy od času 0 po poradí ako boli najvýbušnejšie. V prípade, že v nejakom čase neexistuje žiadna kauza, vypíšte `nic a b`, kde `a b` sú časy odkedy a dokedy sa nič nedeje.

Príklad vstupu:

```
3
medvede 1 10 3
hasici 3 6 5
ukradli 4 7 6
```

Príklad výstupu:

```
nic 0 0
medvede 1 2
hasici 3 3
ukradli 4 7
medvede 8 10
```

Príklad vstupu:

```
4
medvede 2 4 5
diplomovka 2 4 10
dane 7 8 6
uhorky 7 8 1
```

Príklad výstupu:

```
nic 0 1
diplomovka 2 4
nic 5 6
dane 7 8
```

4. [*20 bodov*] **Reprezentatívna výška v škole (programátorská úloha).** V neďalekej základnej škole pre deti diplomatov je k detí. Diplomati sa často sťahujú, každý deň preto ubudne zo školy dieťa, ktoré je v škole najdlhšie, a pribudne jedno nové dieťa. Aby sme vedeli, či nám deti diplomatov neprerastajú cez hlavu, je potrebné sledovať reprezentatívnu výšku detí v škole, a preto pani riaditeľka každý deň na štatistický úrad musí nahlasovať výšku $\lfloor (k-1)/2 \rfloor$-tého najnižšieho dieťaťa. Prvých k dní deti do školy len pribúdajú a prvá štatistika sa hlási v deň $k-1$. Napíšte program, ktorý pomôže pani riaditeľke s týmito štatistickými reportami.

Vstup. Na vstupe sú dva riadky. Na prvom je dvojica medzerou oddelených celých čísel n, k. Platí, že $2 \leq k \leq n \leq 1\,000\,000$, a na druhom postupnosť n medzerou oddelených celých čísel $a_0, \ldots, a_{n-1}$ $(0 \leq a_i < 10\,000\,000)$. Číslo a_i je výška i-teho dieťaťa. Môžete predpokladať, že všetky deti sú rozdielne vysoké, teda že $\forall i, j : i \neq j \rightarrow a_i \neq a_j$.

Výstup. Na výstup vypíšte jediný riadok obsahujúci $n - k + 1$ čísel. i-te číslo v riadku znamená reprezentatívnu výšku detí v deň $k + i$. Dajte si pozor a *ne*vypíšte za posledným číslom medzeru, vypíšte iba znak nového riadku.

Príklad vstupu:

```
5 3
0 1 2 3 4
```

Príklad výstupu:

```
1 2 3
```

Štatistika sa robí z týchto výšok detí:

a) `0 1 2` → odpoveď 1

b) `1 2 3` → odpoveď 2

c) `2 3 4` → odpoveď 3

Príklad vstupu:

```
4 2
42 10 22 95
```

Príklad výstupu:

```
10 10 22
```

Všeobecné pokyny

Čítajte poriadne všetko, čo je napísané čierne na bielom. Riešenia, ktoré odovzdáte, musia byť vaše vlastné. Neopisujte a nesnažte sa nájsť riešenia v literatúre alebo na internete! Ak ste v priebehu riešenia úlohy diskutovali so spolužiakmi alebo s ChatGPT, napíšte vlastné riešenie s časovým odstupom a v riešení uveďte s kým/čím ste diskutovali (bližšie informácie na stránke predmetu).

60 bodov je považovaných za 100%. Pokiaľ z úlohy získate viac ako 60 bodov, tieto body navyše sú bonusové a zarátavajú sa vám do celkovej známky. **Riešenie písomnej úlohy odovzdané v anglickom jazyku aspoň za polovicu bodov získa ďalšie 2 bonusové body navyše.**

Písomné úlohy. Písomné úlohy odovzdávajte na adrese `https://vektor.fmph.uniba.sk` ako PDF súbory v stanovenom termíne. **Každý príklad odovzdajte v osobitnom PDF súbore.** Na neskoro odovzdané riešenia sa nebude prihliadať.

Píšte riešenia takým spôsobom, aby obsahovali všetku potrebnú informáciu na pochopenie vášho riešenia, ale súčasne aby boli stručné a ľahko pochopiteľné. Všetky tvrdenia je potrebné zdôvodniť (a to aj v prípade, že to nie je explicitne napísané v zadaní).

Ak sa v zadaní požaduje vyriešenie algoritmickej úlohy, odovzdajte najlepší algoritmus, aký viete navrhnúť. Základným kritériom na hodnotenie bude *správnosť algoritmu*, druhým kritériom bude jeho *časová, prípadne pamäťová zložitosť*. Správny ale pomalý algoritmus dostane podstatne viac bodov ako algoritmus, ktorý je síce rýchly, ale nedá správnu odpoveď na každý vstup. Neefektívne algoritmy spĺňajúce podmienky zadania dostanú cca 50% bodov. Súčasťou vášho riešenia musia byť nasledujúce časti:

- Najprv popíšte hlavnú myšlienku algoritmu.
- Vyjadrite algoritmus formou pseudokódu.
- Ak to nie je zrejmé na prvý pohľad, ukážte že váš algoritmus je správny.
- Nezabudnite na analýzu zložitosti algoritmu.

Ak nie je povedané inak, logaritmy majú základ 2.

Programátorské úlohy. Pri programátorských úlohách je úlohou odovzdať len funkčný program, nie je vyžadované písomné riešenie. Riešenie odovzdávate priamo na adrese `https://vektor.fmph.uniba.sk`, kde bude okamžite otestované na niekoľkých sadách vstupov a dozviete sa, koľko bodov získalo. Body získate, keď všetky vstupy z danej sady vyriešite správne v časovom limite. Riešenie môžete odovzdávať aj viackrát, hodnotí sa najlepšie riešenie odovzdané v stanovenom termíne, počet pokusov nie je obmedzený.

Domácu úlohu môžete odovzdať v ktoromkoľvek podporovanom programovacom jazyku. Upozorňujeme však, že vzorové riešenia sú ladené pre C++ a Python. Pri použití iných jazykov negarantujeme, že vaše riešenie dosiahne plný počet bodov. Podrobnosti o tom, ako má váš program vyzerať (vrátane zoznamu podporovaných programovacích jazykov), nájdete na stránke `https://judge.ksp.sk/docs/co-odovzdavat/`.