Pravidlá

Riešenia tejto písomnej skúšky je potrebné odovzdať do **štvrtku 1. 2. 2018, 10:00** e-mailom na `misof@ksp.sk`. Ak odovzdávate viacero súborov, zozipujte ich prosím. Počas riešenia priebežne **sledujte oznamy** na webstránke predmetu.

Smiete využívať ľubovoľné neživé zdroje informácií ktoré existovali v okamihu začiatku písomky. Pochopiteľne, vrátane všetkého zverejneného na stránke predmetu. Až do deadline je zakázané akýmkoľvek spôsobom diskutovať o úlohách s kýmkoľvek živým. Použité externé zdroje dostatočne **adekvátne citujte**. Čo viete odcitovať, netreba rozpisovať. Ak ste pri riešení úlohy písali program, odovzdajte aj ten.

Do hodnotenia sa vám započíta **5 najlepšie vyriešených** úloh. Zapísaniu známky môže predchádzať rozhovor o niektorých úlohách ktoré ste riešili.

0 Nultá podúloha

Do textu riešenia nezabudnite prosím vyplniť dva údaje: celé meno a **odhadovaný čistý čas** (v hodinách) strávený riešením úloh. Tento údaj **nebude mať vplyv na hodnotenie**, chcem ho kvôli lepšej kalibrácii úloh v budúcnosti.

1 Podreťazce

V tejto úlohe používame abecedu $\Sigma = \{a, b, c, d, r\}$.
V závislosti od n, čoho je viac: reťazcov dĺžky n ktoré **obsahujú** (súvislý) podreťazec $abracadabra$, alebo reťazcov, ktoré ho neobsahujú?
Asymptoticky, koľko je ktorých z nich?

2 Kocka so zlou jednotkou

Hráme sa s obyčajnou kockou. Snažíme sa hodiť aspoň raz každú z hodnôt 2 až 6. Ale vždy, keď padne 1, sa celý stav hry zresetuje.
Príklad priebehu hry: 2, 3, 1, 4, 4, 5, 6, 3, 4, 2.
Bude nás zaujímať hodnota e: priemerný počet hodov potrebný na vyhratie takejto hry. Inými slovami, e je stredná hodnota počtu hodov potrebných na to, aby aspoň raz padli čísla 2-6 bez toho, aby medzi nimi padlo číslo 1.
(2 body) Ľubovoľným spôsobom rozumne odhadnite e.
(4 body) Navrhnite ľubovoľný algoritmus, ktorý nám umožní v rozumnom čase e určiť na 9 platných cifier.
(8 bodov) Navrhnite polynomiálny algoritmus, ktorým vieme vypočítať presnú hodnotu e. (Vzorec v uzavretom tvare je tiež polynomiálny algoritmus. Nik ale netvrdí, že taký vzorec existuje.)
(10 bodov) Spravte 8-bodové riešenie a následne ním naozaj vypočítajte presnú hodnotu e. (Stačí ako floating-point číslo, netreba v uzavretom tvare.)

3 Pizza

V miestnosti je n krabíc pizze a dvaja hráči. V každej krabici je pizza rozkrájaná na niekoľko kúskov. Hráči striedavo ťahajú. Ťah vyzerá tak, že hráč z jednej krabice zoberie buď jeden alebo dva kúsky pizze a tie zje. (Môže zobrať ľubovoľné dva kúsky, musia však byť z tej istej krabice.) Hru prehráva hráč, ktorý nemá platný ťah – teda vyhráva ten, kto zožerie posledný kúsok pizze.
Navrhnite čo najefektívnejší algoritmus, ktorý načíta počty kúskov pizze v jednotlivých krabiciach a zistí: 1. či je hráč na ťahu vo vyhrávajúcej pozícií, 2. ak áno, aký nasledujúci ťah má spraviť. (Stačí nájsť ľubovoľný jeden, ak je viac možností.)

4 Krajina neplatičov

Na vstupe sú dané dva zoznamy ľudí: zoznam všetkých ľudí v krajine a zoznam neplatičov – ľudí, ktorí majú za minulý kalendárny rok nedoplatky. (Druhý zoznam je podmnožinou prvého. Pre jednoduchosť predpokladáme, že všetci ľudia majú jednoznačné identifikátory.)
Chceme mať databázu neplatičov. Presnejšie, všetkých neplatičov chceme mať uložených vo vrcholoch binárneho vyhľadávacieho stromu (BST).
V budúcnosti si ľudia budú v našej databáze vyhľadávať. Predpokladáme, že vyhľadávanie každého človeka v krajine je rovnako pravdepodobné. Čas vyhľadávania človeka je priamo úmerný počtu vrcholov stromu, na ktoré sa pozrieme.
Navrhnite čo najefektívnejší algoritmus, ktorý načíta oba zoznamy a z nich zostrojí ten strom s databázou, ktorý má spomedzi všetkých možných stromov najmenší priemerný čas vyhľadávania.
Príklad: Ak sú v krajine ľudia A, B, C, z čoho A a B sú neplatiči, optimálny strom má v koreni B a v jeho ľavom synovi A.

5 Krajina tunelov

Švajčiarsko je plné tunelov, a tak nie je ľahké dostať kamión či žeriav z miesta A na miesto B – tunely sú rôzne vysoké a nie každé vozidlo sa do každého tunelu zmestí.
Daná je matica, ktorá pre každé dve mestá udáva buď hodnotu 0 (ak nie sú spojené priamou cestou), alebo maximálnu výšku v cm vozidla, ktoré vie prejsť po priamej ceste medzi nimi.
Navrhnite čo najefektívnejší algoritmus, ktorý pre každú dvojicu miest zistí najvyššiu výšku vozidla, ktoré vie prejsť z jedného z nich do druhého (či už priamo alebo cez iné mestá).

6 Kde je Waldo?

Dané sú dve bitmapy: veľká bitmapa ukazujúca, ako Waldo vyzerá, a obrovská bitmapa ukazujúca nejakú komplikovanú scénu. Waldo je ukrytý niekde v scéne. Vyzerá presne ako na svojej bitmape, môže však byť sčasti niečím zakrytý.
Pre jednoduchosť môžete predpokladať, že oba obrázky majú tú istú 256-farebnú paletu. (A prípadne že jedna z tých farieb je priesvitná a tá je použitá len na Waldovej bitmape na jeho okolie.)
Navrhnite čo najefektívnejší algoritmus, ktorý na obrovskej bitmape nájde miesto, kde sa Waldo ukryl. (Za algoritmus, ktorého časová zložitosť je priamo úmerná súčinu veľkostí bitmáp, skoro žiadne body nedostanete.)

7 Náhodné stromy

Číslo n je párne a väčšie ako 100.
Xénia zobrala náhodnú permutáciu čísel 1 až n a jej prvky postupne vložila do (pôvodne prázdneho, nevyvažovaného) BST. Yvona rozdelila čísla na malé (1 až $n/2$) a veľké (ostatné). Potom spravila náhodnú permutáciu malých a náhodnú permutáciu veľkých. Na záver postupne vložila všetky čísla do BST, pričom na striedačku vkladala jedno malé (v poradí podľa ich permutácie) a jedno veľké.
Anička hovorí: „Očakávaný čas potrebný na výrobu Xéniinho stromu je rovnaký ako očakávaný čas potrebný na výrobu Yvoninho stromu.“
Boris hovorí: „Očakávaná maximálna hĺbka Xéniinho stromu je rovnaká ako očakávaná maximálna hĺbka Yvoninho stromu.“
Cyril hovorí: „Boris sa mýli, očakávaná maximálna hĺbka Xéniinho stromu je presne o 1 menšia ako očakávaná maximálna hĺbka Yvoninho stromu.“
Ktoré výroky sú pravdivé a prečo?