Toto sú návodné úlohy k domácemu kolu 41. ročníka Olympiády v informatike. Ide teda o sadu ľahších úloh, ktoré tematicky súvisia so súťažnými úlohami. Riešenie týchto úloh môže byť dobrou prípravou na riešenie súťažných úloh. Za riešenia týchto návodných úloh nie sú žiadne body do súťaže. O ich riešení môžete ľubovoľne diskutovať – so svojim učiteľom, inými súťažiacimi aj kýmkoľvek iným.

Návodné úlohy k domácemu kolu OI: kategória A

A-I-1 Semafory

1. Uvažujme kalendárny rok, ktorý začal v pondelok. Napíšte program, ktorý načíta poradové číslo dňa v roku (napr. 1 je prvý január, 34 je tretí február a podobne) a vypočíta, či v ten deň je alebo nie je víkend.

   Potom upravte vyššie uvedený program tak, aby fungoval pre rok 2025 (prvý január 2025 bola streda) a prípadne aj tak, aby bral ako vstup dátum – teda číslo dňa a číslo mesiaca.

2. Ako by vyzeralo riešenie súťažnej úlohy „na priamke“? Inými slovami, predpokladajte, že existuje presne $n-1$ ciest, pričom i-ta z nich vedie od križovatky i ku križovatke $i+1$. A samozrejme, na vstupe na každú cestu je stále, rovnako ako v súťažnej úlohe, semafor.

3. Súťažná úloha má niekoľko rôznych sád vstupov. Ku každej sú v zadaní uvedené dodatočné obmedzenia.

   Ku čo najviac sadám (ku každej zvlášť) skúste vymyslieť jednoduchšie riešenie, ktoré by vyriešilo takéto vstupy.

   Ako by ste napríklad súťažnú úlohu vyriešili, keby ste mali zaručené, že na každom semafore stále svieti zelená?

   (Takéto riešenie môžete následne naprogramovať, odovzdať a dostať tak nejaké body.)

4. Ak už viete riešiť súťažnú úlohu pre semafory, ktoré majú stále zelenú, vyriešte si následne nasledovnú verziu:

   Vstup vyzerá presne rovnako ako v súťažnej úlohe, ale v dopravnom dispečingu sedí váš úhlavný nepriateľ. Ten vždy, keď vidí, že sa blížite ku križovatke, posunie cykly všetkých semaforov na cestách vedúcich z nej ďalej tak, aby na nich v okamihu, keď do križovatky prídete, práve naskočila červená. Na úplne každom semafore teda budete cestou musieť najdlhšiu možnú dobu čakať, kým sa zmení na zelenú.

   Ako nájsť najrýchlejšiu cestu zo štartu do cieľa v takejto situácii?

A-I-2 Pokazený robot

V prvých štyroch návodných úlohách máme vždy ten istý vstup: daná je postupnosť $n \leq 200\,000$ celých čísel (môžu byť aj záporné) a číslo k ($1 \leq k \leq n$). Všimnite si, že existuje $n-k+1$ spôsobov, ako v tejto postupnosti vyznačiť súvislý úsek dĺžky k.

1. Navrhnite čo najefektívnejší algoritmus, ktorý vypočíta a vypíše $n-k+1$ čísel: ku každému možnému úseku dĺžky k jeho súčet.

2. Navrhnite čo najefektívnejší algoritmus, ktorý vypočíta a vypíše $n-k+1$ čísel: ku každému možnému úseku dĺžky k jeho maximum.

3. Navrhnite čo najefektívnejší algoritmus, ktorý vypočíta a vypíše $n-k+1$ čísel: ku každému možnému úseku dĺžky k počet kladných čísel, ktoré obsahuje.

4. Predpokladajme navyše $k \geq 3$. Navrhnite čo najefektívnejší algoritmus, ktorý vypočíta a vypíše $n-k+1$ čísel: ku každému možnému úseku dĺžky k súčet troch najväčších prvkov v ňom.

5. Upravme súťažnú úlohu tak, že namiesto p po sebe idúcich príkazov mohol robot vykonať **ľubovoľných** p spomedzi zadaných n príkazov. Naďalej platí, že nanajvýš k z nich mohol vykonať opačne. Navrhnite efektívny algoritmus, ktorý pre takto upravené zadanie vypočíta, ako najďalej od svojej začiatočnej polohy mohol robot skončiť.

A-I-3 Piškóty

1. Na stole je v rade 23 pohárov, ktoré sú na striedačku otočené normálne a hore nohami. (Prvý je otočený normálne, druhý hore nohami, tretí zase normálne, a tak ďalej.)

   Chceli by sme, aby boli všetky poháre otočené normálne. Dovolený máme ale len jeden typ zmeny: môžeme každou rukou chytiť jeden pohár a potom oba naraz otočiť hore nohami. Vieme dosiahnuť náš cieľ? Ako áno / prečo nie?

2. Vedela by predchádzajúcu úlohu vyriešiť chobotnica, ktorá v každom kroku použije svojich osem chápadiel na otočenie ôsmich pohárov?

   Vedela by ju vyriešiť opica, ktorá v každom kroku použije dve ruky a chvost?

3. Navrhnite čo najefektívnejší algoritmus, ktorý bude riešiť všeobecnejšiu verziu tejto úlohy: načíta zo vstupu začiatočné otočenia 23 pohárov (napr. postupnosť 23 núl a jednotiek) a potom slovo „človek“, „chobotnica“ alebo „opica“ a vypíše buď návod na postupné otáčanie pohárov (ak nejaký postup existuje) alebo správu, že to nejde (ak nie).

4. Janko postavil na klasickú šachovnicu dvoch kráľov: jedného do jedného rohu (políčko a1) a druhého do úplne protiľahlého rohu (h8). Potom si doniesol sadu domín. Každé domino má rovnakú veľkosť ako dve políčka šachovnice. Janko sa dlho neúspešne snažil celý zvyšok šachovnice (teda zvyšných 62 políčok) pokryť dominami tak, aby každé políčko bolo zakryté práve jedným dominom. Rozmyslite si, prečo sa mu to nikdy nemohlo podariť.

   Nasledujúcu úlohu si prečítajte až po vyriešení tejto.

5. Janko samozrejme (rovnako ako dúfam aj vy) časom odhalil, že tajomstvo za jeho neúspechom je v šachovnicovom ofarbení políčok. Obe políčka, kde stáli králi, mali rovnakú farbu – na klasickej šachovnici sú obe čierne. To ale znamenalo, že mu na pokrytie dominami ostalo 32 bielych a len 30 čiernych políčok. A to samozrejme nemôže ísť všetko pokryť dominami, keďže každé domino pokryje jedno biele a jedno čierne políčko.

   Janko sa prišiel svojim objavom pochváliť Peťke. A potom sa ešte trochu zamyslel a povedal: „Vlastne by to malo platiť aj všeobecnejšie. Keď si na štvorčekovom papieri vyznačíš ľubovoľnú súvislú oblasť, tak si môžeš následne jej políčka ofarbiť ako šachovnicu. Ak dostaneš rôzny počet čiernych a bielych políčok, tak sa tá oblasť dominami pokryť nedá, a ak dostaneš rovnaký, tak sa dá.“

   „Hm, to ti podľa mňa nebude úplne vždy fungovať,“ nebola presvedčená Peťka.

   Rozhodnite, kto z nich má pravdu, a svoje tvrdenie príslušne zdôvodnite.

6. V rade je n jamiek. Sú za radom očíslované od 1 po n. V jamke i je g_i guličiek. V každom kroku si môžeme vybrať dve *susedné* jamky a buď do každej jednu guličku pridať, alebo (ale len ak práve ani jedna nie je prázdna!) z oboch po jednej guličke odobrať.

   Navrhnite algoritmus, ktorý pre danú postupnosť g_i rozhodne, či vieme dosiahnuť, aby boli všetky jamky prázdne.

   (Dobrá rada: Možno pomôže predstaviť si, že v jamkách s párnym číslom používame modré a v jamkách s nepárnym číslom červené guličky.)

A-I-4 Nechaj to na solver

1. V študijnom texte sme si v časti „Príklad: sudoku“ ukázali, ako namodelovať sudoku pomocou $9 \times 9 \times 9$ binárnych premenných. Teraz by sme chceli pridať do takéhoto programu 9×9 nových (všeobecných celočíselných) premenných $v_{i,j}$ tak, aby v riešení platilo, že $v_{i,j}$ je hodnota cifry, ktorá sa v riešení nachádza na políčku (i, j).

   Aké obmedzenia máme do programu pridať aby sme toto dosiahli?

2. V príklade „Príklad: sudoku“ sme mali obmedzenia s logickým významom „V každom riadku/stĺpci/štvorci sa každé číslo nachádza práve raz.“

   Ak by sme namiesto slova *práve* použili slovo *najviac*, dostali by sme namiesto rovností nerovnosti: napr. pre i-ty riadok a hodnotu k by sme mali podmienku $x_{i,1,k} + x_{i,2,k} + \cdots + x_{i,9,k} \leq 1$.

   Zmenilo by sa tým niečo podstatné? Presnejšie, keď takto zmeníme ILP zodpovedajúci konkrétnemu sudoku, bude mať stále jediné platné riešenie zodpovedajúce riešeniu pôvodného sudoku?

3. Opäť sa pozrime na ILP, ktorý sme vyrobili v príklade „Príklad: sudoku“. Je v tomto programe skutočne potrebné mať obmedzenia s logickým významom „Na každom políčku je práve jedno číslo“? Čo by sa stalo, keby sme túto časť programu vynechali? Fungoval by naďalej či nie?

4. *Killer sudoku* (https://en.wikipedia.org/wiki/Killer_sudoku) je variant sudoku, pri ktorom sú v zadaní vyznačené oblasti, nazývané tiež *klietky*. Každá klietka má v sebe napísané malé číslo a platia pre ňu dve pravidlá: žiadne dve cifry v klietke nesmú byť rovnaké a súčet cifier v klietke musí byť presne rovný v nej uvedenému malému číslu.

   a) Rozmyslite si, ako vhodne reprezentovať killer sudoku vo formáte, ktorý sa bude dobre spracúvať na počítači.

   b) Prepíšte ukážkové killer sudoku z Wikipédie do vami zvoleného formátu.

   c) Vysvetlite, ako vieme killer sudoku modelovať ako ILP.

   d) Napíšte program, ktorý vygeneruje ILP zodpovedajúci killer sudoku z Wikipédie a následne pomocou solveru nájdite jeho riešenie a overte si, že vám vyšlo správne.

5. V pivnici je n predmetov, predmet i má hmotnosť v_i gramov. Chceme toho z pivnice vyniesť čo najviac von – presnejšie, chceme vyniesť veci s najväčšou možnou celkovou hmotnosťou. Problém je samozrejme v tom, že máme len obmedzenú nosnosť: nevieme odniesť viac ako ℓ gramov.

   Zostrojte ILP, ktorého optimálne riešenie nám povie, koľko najviac gramov vecí vieme vyniesť jednou cestou z pivnice a ktoré veci máme zobrať, aby sme to dosiahli.