

1. Mi az alábbi állításoknak a tagadása? *(Két állítás akkor tagadása egymásnak, ha a két állítás közül minden esetben pontosan az egyik igaz.)* Próbáljuk úgy megfogalmazni a tagadásokat, hogy ne szerepeljen bennük tagadószó.
  - (a) Minden szerdán van algel előadás.
  - (b) Minden olyan hallgató, aki jár algel gyakorlatra, teljesíti a tárgyat.
  - (c) Minden olyan 17 lábú zsiráf, aki jár algel gyakorlatra, az teljesíti a tárgyat. *(Igaz ez?)*
  - (d) Van olyan hallgató, aki sokat tanul, de nem teljesíti a tárgyat.
  - (e) Mindenki, aki teljesíti a tárgyat, sokat tanult.
2. Tudjuk, hogy minden hömpörő surjancs. Mondjuk meg minden alábbi állításra, hogy biztosan igaz, lehetséges, vagy biztosan hamis! *(Ha nehéz a feladat, akkor legyen a hömpörő=kertitörpe és surjancs=szobor.)*
  - (a) Tudjuk valamiről, hogy nem hömpörő. Azt állítom, hogy ez surjancs.
  - (b) Tudjuk valamiről, hogy hömpörő. Azt állítom, hogy ez nem surjancs.
  - (c) Tudjuk valamiről, hogy nem surjancs. Azt állítom, hogy ez hömpörő.
  - (d) Tudjuk valamiről, hogy nem surjancs. Azt állítom, hogy ez nem hömpörő.
  - (e) Tudjuk valamiről, hogy surjancs. Azt állítom, hogy ez nem hömpörő.
3. Mi lehet a  $T(n)$  függvény, ha teljesül  $T(1) = 2$  és  $T(n) = 3 \cdot T(n - 1) + 1$  minden  $n \geq 2$  esetén?
4. Jelölje egy algoritmus maximális lépésszámát az  $n$  hosszú bemeneteken  $L(n)$ . Azt tudjuk, hogy minden  $n > 3$  egész számra  $L(n) \leq L(n - 1) + \frac{n}{2}$  teljesül, és hogy  $L(3) = 3$ . Milyen felső becslést adhatunk ez alapján  $L(n)$ -re?
5. Tegyük fel, hogy van egy számítógépes programunk, ami egy  $k$  méretű feladaton a jelenlegi gépünkön 1 nap alatt fut le. Beszereztünk egy százszor gyorsabb számítógépet. Ugyanazon programmal mekkora feladatot lehet az új gépen egy nap alatt megoldani, ha a program lépésszáma  $n$  méretű feladat esetén
  - (a)  $n$ -nel arányos,      (b)  $n^3$ -bel arányos,      (c)  $2^n$ -nel arányos?
6. Egy  $f$  fokú létrán bizonyos fokok annyira rozogák, hogy ha rálépünk, leszakadnak. Szerencsére tudjuk hogy melyik fokok ilyenek, hova nem szabad lépniük. Egy lépéssel legfeljebb 3 fokot tudunk lépni. Adjunk algoritmust ami meghatározza, hogy a létra aljától fel tudunk-e jutni a létra legfelső fokára! *(Feltehető, hogy a legfelső fokra rá szabad lépni.)* Az algoritmus lépésszáma legyen  $c \cdot f$ , ahol  $c$  valami fix konstans. Hogyan kell módosítani az algoritmust, hogy azt is kiszámolja, hogy hányféleképpen lehet feljutni a legfelső fokra?
7. Adott  $n$  chip, melyek képesek egymás tesztelésére a következő módon: ha összekapcsolunk két chipet, mindkét chip nyilatkozik a másíkról, hogy hibásnak találta-e. Egy hibátlan chip korrektül felismeri, hogy a másik hibás-e, míg egy hibás chip akármilyen választ adhat. Tegyük fel, hogy a chipek több, mint a fele korrekt. Adjunk algoritmust, mely  $n$ -nél kevesebb fenti tesztet használva kikeres egy jó chipet.
8. Egy tanteremben fel van szerelve egy  $n \times n$ -es tábla, melyen  $n^2$  villanykörte helyezkedik el. A tábla minden egyes sorához illetve oszlopához tartozik egy-egy nyomógomb, mellyel a megfelelő sorban (oszlopban) található  $n$  darab villanykörte állapotát egyszerre lehet átváltoztatni az ellenkezőjére. *(Egy gombnyomásra az adott sorban illetve oszlopban égő körték elalszanak, az alvók pedig kigyulladnak.)* A szünet kezdetekor az összes körte leoltott állapotban van. Szünetben a nebulók össze-vissza nyomogatják a gombokat. Hány kapcsolással tudja a tanár visszaállítani az eredeti állapotot? *(A gombok egyállapotúak, azaz nem látszik rajtuk, hogy megnyomták-e őket vagy sem.)*
9. Egy  $2 \times n$ -es sakktábla mezőin  $n$  piros és  $n - 1$  kék négyzetet helyezünk el. Ezeket olyan módon akarjuk átrendezni, hogy a felső sorban piros, az alsóban kék négyzetek legyenek, s a bal alsó sarok maradjon üres. Ehhez egy-egy lépés során az üres mezőre tolhatjuk valamelyik szomszédját. Bizonyítsuk be, hogy
  - (a) van olyan algoritmus, ami ezt megoldja  $c \cdot n^2$  lépéssel, ahol  $c$  valamilyen fix konstans
  - (b) létezik olyan  $d$  konstans, hogy minden algoritmus, ami ezt megoldja, szerencsétlen inputon használ legalább  $d \cdot n^2$  lépést.