1. Adjon időkorlátot az alábbi feladatokban megkonstruált Turing-gépekhez!

- Adjon 1-szalagos determinisztikus Turing-gépet a $\{\texttt{0}^{2^m} \mid m \geq 0\}$ nyelvhez!

  Megoldás: Az első `0`-t átírjuk `1`-re, ez jelöli majd az input elején az első `0`-t. Jobbra haladva minden második `0`-t átírjuk `x`-re. Ha az utolsót nem írtuk át, akkor páratlan hosszú volt az input, elutasítjuk. Ha átírtuk, akkor páros hosszú volt és a megmaradt `0`-k száma feleződött. Visszamegyünk az elejére és megint minden második `0`-t átírjuk `x`-re. Az útközben talált `x`-eken csak jobbra lépünk. Ezt addig ismételjük, amíg lehet. Ha végül nem marad `0`, csak az elején az `1`-es, akkor elfogadjuk. *Időkorlát:* A bemenet mérete $n = 2^m$. Az első fázisban az input végéig megyünk, majd vissza az elejére, ez $O(n)$ lépés. Ezt annyiszor kell ismételni, ahányszor 2^m-et el kell osztani 2-vel, hogy 1-et kapjunk. Ez $\log_2(2^m) = m = \log_2(n)$. Tehát az időkorlát $O(n \log_2(n))$.

- Adjon 2-szalagos nemdeterminisztikus Turing-gépet a $\{ww \mid w \in \{\texttt{0},\texttt{1}\}^*\}$ nyelvhez!

  Megoldás: Először megjelöljük a 2. szalag elejét, majd lemásoljuk az input első néhány karakterét a 2. szalagra. Minden karakter másolása után lehetőség van a következő karakter másolására vagy a másolás befejezésére és átlépésre egy új állapotba nemdeterminisztikus módon. Az új állapotban a 2. szalag fejét az elejére visszük. Ezután mindkét fejjel jobbra haladva a szó első felét a 2. szalagon karakterenként össze tudjuk hasonlítani az 1. szalagon a szó második felével. *Időkorlát:* az első fej minden lépésben jobbra lép vagy helyben marad és akkor a második fej balra haladva legfeljebb n lépést tesz, ezért a NTG bármely számítási úton $O(n)$ lépést végez.

- Adjon 2-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{\texttt{0},\texttt{1}\}^*\}$ nyelvhez!

  Megoldás: 2 szalagos gépet készítünk. Először megjelöljük a 2. szalag elejét, majd lemásoljuk az egész inputot a 2. szalagra. Visszafelé haladva az 1. szalagon csak minden második lépésben mozdítjuk a fejet balra, így amikor a 2. szalagon visszaérünk az elejére, akkor az 1. szalagon a fej középen lesz. Ezután jobbra haladva a 2. szalagon a szó első felét karakterenként össze tudjuk hasonlítani az 1. szalagon a szó második felével. *Időkorlát:* Mindhárom fázisban legfeljebb $n + 1$ lépést végez a TG, tehát az időkorlát $O(n)$.

- Adjon 1-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{\texttt{0},\texttt{1}\}^*\}$ nyelvhez!

  Megoldás: Meg kell keresnünk valahogy az input közepét: ehhez a 0-kat x-re, az 1-eket y-ra változtatjuk a következő módon. Átírjuk az első karaktert x-re ill. y-ra, majd az utolsó karaktert is. Mindig visszamegyünk az első 0 vagy 1 karakterre, azt is átírjuk, majd az utolsó ilyet is. A legelső karaktert meg akarjuk különböztetni a többitől, hogy fel tudjuk ismerni, hogy a szalag elején vagyunk, ezért a legelső karaktert x helyett a-ra, y helyett b-re változtatjuk. Ha nem sikerül mindent átírni, akkor az input hossza páratlan volt, elutasítjuk. Ha sikerült, akkor az utolsó átírt karakter épp a második w első karaktere, ezt írjuk át z-re, de menjünk különböző állapotba, ha ott 0 ill. 1 volt. Menjünk a szalag elejére, ott el tudjuk dönteni, hogy a olvasott karakter megegyezik-e a megjegyzett karakterrel, ami a második rész első karaktere. Ha nem, elutasítjuk az inputot. Ha igen, akkor írjuk át ezt is z-re, majd oda vissza ingázva a két rész között, összehasonlítjuk a két részt. Ha minden stimmel, akkor elfogadjuk az inputot. *Időkorlát:* Az input közepének megkereséséhez $n/2$ alkalommal megyünk jobbra, majd balra, mindig legfeljebb n távolságra, ez $O(n^2)$ lépés. Az összehasonlítás során pedig $n/2$ alkalommal megyünk jobbra, majd balra, mindig legfeljebb $n/2$ távolságra, ez is $O(n^2)$ lépés. Tehát az időkorlát $O(n^2)$.

- Adjon 2-szalagos nemdeterminisztikus Turing-gépet az $\{1^m \mid m \text{ összetett szám}\}$ nyelvhez!

  Megoldás: Először megjelöljük a 2. szalag elejét, majd néhány x-et írunk a 2. szalagra. Minden karakter írása után lehetőség van még egy x írására vagy az írás befejezésére és átlépésre egy új állapotba nemdeterminisztikus módon. Az új állapotban a 2. szalag fejét az elejére visszük. Ezután mindkét fejjel jobbra haladva, amíg x-et olvasunk a 2. szalagon és 1-et az 1. szalagon, addig jobbra lépünk mindkét szalagon. Ha második szalagon egy $*$-hoz érünk és az 1. fej 1-et olvas, akkor a 2. fejet az elejére mozgatjuk, és ismételjük az előzőeket. Ha második szalagon egy $*$-hoz érünk és ekkor az 1. fej is $*$-ot olvas, akkor az egyesek száma osztható volt az x-ek számával, ezért elfogadjuk az inputot. *Időkorlát:* Vigyázat, ez a NTG így nem időkorlátos, mert az x-ek írása az elején bármeddig eltarthat. Ha azonban úgy módosítjuk, a fenti NTG-t, hogy legfeljebb annyi x-et írjon nemdeterminisztikusan, mint az input hossza, akkor már időkorlátos lesz. Az x-ek írása és az oszthatóság ellenőrzése is minden számítási úton $O(n)$ lépés.

2. Álljon az L nyelv azokból az (n, m) párokból, amelyekben n és m egy-egy pozitív egész szám bináris alakja, és ez a két szám relatív prím. Igaz-e, hogy $L \in \mathrm{P}$?

   Megoldás: Itt a bemenet hossza $\Theta(\log n + \log m)$, de az euklideszi algoritmus, amivel két szám legnagyobb közös osztója meghatározható, ebben is polinom idejű algoritmus.

3. Bizonyítsa be, hogy nincs olyan polinom időkorlátos Turing-gép, amelynek bemenete egy pozitív egész szám k bináris alakban, kimenete pedig 2^k bináris alakban!

   Megoldás: A bemenet mérete $n = \log_2(k)$, a kimenet mérete viszont $\log_2(2^k) = k = 2^n$, vagyis exponenciális. Mivel tudjuk, hogy $2^n \notin O(n^k)$, ez tehát nem polinom. A kimenetről könnyen látható, hogy egy `1`után k darab `0`következik. Ennek a kimeneti szalagra írásához a Turing-gép mindenképp kénytelen k lépést végezni, ami viszont nem polinomiális az input méretében.

4. Adjon polinom időkorlátos Turing-gépet, amelynek bemenete két pozitív egész szám k és m bináris alakban, kimenete pedig $2^k \mod m$ (vagyis 2^k maradéka m-el osztva) szintén bináris alakban!

   Megoldás: Először is vegyük észre, hogy itt nincs probléma a kimenet méretével, hiszen a kimenet egy m-nél kisebb szám, ami tehát nem hosszabb a bemenetnél. Az algoritmus pedig az ismételt négyzetre emelések módszere, ami volt BSZ-ből.

5. Álljon az L nyelv azokból az irányítatlan gráfokból, melyekben nincs kör. Igazolja, hogy $L \in \mathrm{P}$.

   Megoldás: Elegendő azt megmutatni, hogy erre az eldöntési feladatra van polinom idejű algoritmus. A szélességi (vagy mélységi) bejárás alkalmas erre: egy tetszőleges pontból indítjuk és akkor hagyjuk abba, hogy ha vagy minden pontot bejárt vagy talált egy élet, ami nem faél. Az utóbbi esetben van kör a gráfban. Ha viszont minden él faél, akkor a gráf egy erdő, tehát nincs benne kör.

   Az algoritmus lépésszáma v csúcsú gráfokon mátrixos megadás esetén $O(v^2)$, ami lineáris a bemenet $N = v^2$ hosszában.

   Megjegyzés: Ha ezt Turing-géppel akarjuk megvalósítani, akkor több lépés kell. Hiszen míg a mátrixból két pont közötti él létezését eldönteni $O(1)$ lépés, addig a Turing-gép szalagján oda kell menjünk a megfelelő helyre, ami v^2-tel arányos lépést is jelenthet. De az ebből adódó $O(v^4)$ összes lépésszám is polinomiális v-ben (és így N-ben is).

6. Az L nyelv álljon az olyan (G, s, t) hármasokból, ahol G egy irányított gráf, s és t a gráfnak két csúcsa és G-ben van út s-ből t-be. Igazolja, hogy $L \in \mathrm{P}$.

   Megoldás: Egy s-ből induló szélességi bejárással megválaszolható a kérdés. (A bejárás leállhat, ha elérünk t-be vagy ha bejártuk s komponensének összes csúcsát.) Mivel a BFS polinom idejű, ezért $L \in \mathrm{P}$.

7. Legyen $L_1 \in P$ és $L_2 \in P$. Bizonyítsa be, hogy $L_1 \cup L_2 \in P$ is teljesül!

   Megoldás: Legyen M_1 egy $O(n^k)$ időkorlátos Turing-gép, ami az L_1 nyelvet ismeri fel, M_2 pedig egy $O(n^\ell)$ időkorlátos Turing-gép, ami az L_2 nyelvet. Konstruálunk egy új, M_3 TG-et: A w bemeneten M_3 először futtatja M_1-et, majd ha megállt (biztos megáll, hiszen időkorlátos), akkor szintén a w bemeneten futtatja M_2-t (ez is meg fog állni). Ha bármelyik TG elfogadta az inputot, akkor M_3 is elfogadja, különben elutasítja. M_3 időkorlátja nyilván $O(n^k + n^\ell) = O(n^{\max(n,\ell)})$, vagyis polinomiális.

8. Bizonyítsa be az alábbi két nyelvről, hogy NP-beliek! Melyikről tudja belátni, hogy P-ben van?

   (a) G irányítatlan gráfok nyelve, amelyekben van legfeljebb 100 élből álló kör.

   (b) (G, k) párokból álló nyelv, ahol a G irányítatlan gráfban van legfeljebb k élből álló kör.

Megoldás: (a) A tanú tétel alapján NP-ben van, hiszen legyen az L_1 nyelv az olyan (G, y) párok nyelve, ahol G egy irányítatlan gráf és y egy legfeljebb 100 élű kör csúcsai a kör mentén sorban felsorolva. Világos, hogy G pontosan akkor van benne a feladatbeli L nyelvben, ha van hozzá olyan y, hogy $(G, y) \in L_1$. Ennek az y-nak a hossza, ha a körbeli csúcsok sorszámát binárisan felírjuk, $100 \log v \in O(\log v)$, ahol v a G csúcsainak száma, tehát $|y|$ kisebb, mint a G megadásának hossza, ezért belefér a polinom korlátba. Még az kell, hogy $L_1 \in P$. Ez azért igaz, mert az L_1-be tartozáshoz azt kell ellenőrizni, hogy a bemenet y részében legfeljebb 100 csúcs szerepel, ezek mind különbözőek és hogy a szomszédos csúcsok, valamint az első és utolsó csúcs között megy él. (Igazából azt is megengedhetnénk, hogy a felsorolásban legyenek ismétlődő csúcsok, azaz egy zárt élsorozatot adjanak, mert ha ilyen van, akkor legfeljebb 100 élű kör is van.)

(b) Ahhoz, hogy ez NP-ben van, az előző bizonyítást csak kicsit kell módosítani: az y tanú most egy kört alkotó legfeljebb k darab csúcs felsorolása. A bemenetben k binárisan van leírva, azaz a bemenet hossza $\Theta(v^2 + \log k)$. Formálisan nézve egy $k \log v$ hosszú tanú ebben nem polinomiális. De vegyük észre, hogy csak $k \le v$ esetben lehet megoldás, és ekkor $k \log v \le v \log v < v^2$, tehát a tanú valóban polinom hosszú (sőt, lineáris) a bemenet hosszához viszonyítva. Mivel az L_1-be tartozáshoz legfeljebb $k \le v$ csúcs különbözőségét és ennyi él meglétét kell ellenőrizni, ez megoldható polinom időben, azaz $L_1 \in$ P.

Az (a)-ban megadott nyelv világos, hogy P-ben is benne van, hiszen egy v csúcsú gráf esetén a lehetséges 100 élű körök száma nem több, mint v^{100}, ami polinomiális. Adott 100 pontra annak ellenőrzése, hogy az adott sorrendben kört alkotnak konstans sok lépés, tehát minden lehetőség ellenőrzése összesen $O(v^{100})$ és így polinom idő.

A (b)-nél nem alkalmazhatjuk az előző eljárást, mert a lehetőségek száma most v^k nagyságrendű, ami nem polinomiális v-ben és $\log k$-ban, és itt még az a feltevés sem segít, hogy $k \le v$. Ennek ellenére ez a nyelv is P-ben van. Ehhez azt kell észrevenni, hogy egy gráfban egy legrövidebb kör megtalálható polinom sok lépésben.

Vegyünk egy tetszőleges uv élet a gráfban és a BFS segítségével keressük meg a legrövidebb utat a $G - uv$ gráfban u és v között. Világos, hogy ez az út az uv éllel kiegészítve a legrövidebb uv élet tartalmazó kör G-ben. Ha ugyanezt elvégezzük a gráf minden élére, akkor az így megtalált körök közül a legrövidebb a gráfban egy legkevesebb élű kör.

A e darab BFS lépéseinek száma $O(v^4)$, a keletkező körök hosszának meghatározása összesen megoldható $O(v^2)$ lépésben, tehát az egész polinomiális.

9. Igazolja, hogy a

   (a) MAXKLIKK $= \{(G, k) : G$ irányítatlan gráfban van k pontú klikk$\}$ nyelv NP-ben van.

   (b) 5KLIKK $= \{G : G$ irányítatlan gráfban van 5 pontú klikk$\}$ nyelv

   – NP-ben van,

   – P-ben van.

Megoldás: (a) A tanú tételt használjuk: az L_1 nyelv álljon az olyan (G, k, y) hármasokból, ahol G egy irányítatlan gráf, k egy pozitív egész és y olyan k darab csúcs felsorolása, melyek G-ben egy teljes gráfot alkotnak. Ha $(G, k) \in$ MAXKLIKK, akkor $k \le v$, tehát a megfelelő y hossza polinomiális. Azt kell ellenőrizni, hogy y valóban k darab különböző csúcsot tartalmaz, és hogy bármely kettő között megy él (ami kevesebb, mint $k^2 \le v^2$ él ellenőrzését jelenti), és ez megoldható polinom időben. Ha $(G, k) \notin$ MAXKLIKK, akkor nincs olyan y, ami megfelel ennek az ellenőrzésnek.

(b) Az előző bizonyítás a $k = 5$ esetre alkalmazva jó az NP-beliség bizonyításához.

Az, hogy 5KLIKK $\in$ P abból következik, hogy összesen $\binom{v}{5}$ lehetséges 5 pontú klikk van, ezek ellenőrizhetők összesen $O(v^5)$ lépésben.