1. Adjon 1-szalagos determinisztikus Turing-gépet a $\{0^{2^n} \mid n \geq 0\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

   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.

2. Adjon 2-szalagos nemdeterminisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

   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.

3. Adjon 2-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

   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.

4. Adjon 1-szalagos determinisztikus Turing-gépet a $\{ww \mid w \in \{0,1\}^*\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

   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.

5. Adjon 2-szalagos nemdeterminisztikus Turing-gépet az $\{1^m \mid m \text{ összetett szám}\}$ nyelvhez! Nem szükséges precízen megadni az átmeneteket, elegendő a működés elvét leírni.

   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.

6. Vázoljon egy Turing-gépet, ami a diagonális nyelv komplementerét fogadja el. Megáll-e minden bemeneten a kapott Turing-gép?

Megoldás: Először megvizsgáljuk, hogy az input (w) tényleg egy Turing-gép leírása-e. Ha nem, akkor elfogadjuk. Ha igen, akkor elkezdjük szimulálni a megadott gépet a saját kódján futtatva: A 2. szalagra lemásoljuk w-t és azon végzünk minden műveletet, amit az adott gép végezne. A 3. szalagon a szimulált gép aktuális állapotát tartjuk nyilván. Minden lépésben kódban meg van adva, hogy mit kell lépni a 2. és 3. szalagon olvasott adatok alapján. Könnyű olyan Turing-gépet konstruálni, ami bármely bemenetre, így a saját-kódjára is, végtelen ciklusba kerül. Ha fent konstruált Turing-gépnek ez a bemenete, akkor a szimulálás is végtelen ciklusba fog kerülni. Tehát nem fog minden bemenetre megállni.

7. Vázoljon egy Turing-gépet, ami az alábbi nyelvet fogadja el!

$$L = \{w\#x \mid \text{ az } M_w \text{ elfogadja az } x \text{ első öt karakteréből álló szót}\}$$

Megáll-e minden bemeneten a kapott Turing-gép?

Megoldás: Először megvizsgáljuk, hogy az input w része tényleg egy Turing-gép leírása-e. Ha nem, akkor elutasítjuk az $w\#x$ inputot. Ha igen, akkor elkezdjük szimulálni a w-vel megadott M_w gépet az x első öt karakteréből álló y szón: A 2. szalagra lemásoljuk y-t és azon végzünk minden műveletet, amit az M_w gép végezne. A 3. szalagon a szimulált M_w gép aktuális állapotát tartjuk nyilván. Minden lépésben a kódban (átmeneti függvényben) meg van adva, hogy mit kell lépni a 2. és 3. szalagon olvasott adatok alapján. Ha a szimulált M_w gép leáll, akkor mi is leállunk és elfogadunk pontosan akkor, ha M_w elfogadott. Ha tehát $w\#x$ benne volt a nyelvben, azaz M_w elfogadja az x első öt karakteréből álló y szót, akkor mi is le fogunk állni és elfogadjuk $w\#x$-t , azaz az L nyelv szavait elfogadjuk. Ha $w\#x$ nincs benne az L nyelvben, akkor M_w vagy le sem áll az x első öt karakteréből álló szón, de ekkor mi sem állunk állunk le a szimulációval, tehát nem fogadjuk el $w\#x$-t, vagy pedig M_w elutasítva leáll az x első öt karakteréből álló szón, de ekkor mi is elutasítva állunk le a szimulációval, tehát nem fogadjuk el $w\#x$-t. Mivel lehetséges olyan M_w Turing-gépet konstruálni, ami bármely bemenetre végtelen ciklusba kerül, ezért a most vázolt eljárás minden ilyen w-hez tartozó $w\#x$ pár esetén végtelen ciklusba fog kerülni, tehát nem fog minden bemenetre megállni.

8. Bizonyítsa be, hogy nincs olyan Turing-gép, ami minden bemeneten megáll és az alábbi nyelvet fogadja el!

$$L = \{w\#x \mid M_w \text{ egy Turing-gép és } x \in L(M_w)\}$$

Megoldás: Tegyük fel, hogy van ilyen TG, legyen ez M. Belátjuk, hogy ekkor van olyan TG is, ami az L_d diagonális nyelvet eldönti (azaz minden inputon megáll és L_d szavait fogadja el). Ez ellentmondás, hiszen bizonyítottunk, hogy ilyen TG nem létezik.

Legyen M' egy olyan TG, ami ugyanazt csinálja, mint M, csak amikor M elfogadó állapotban áll meg, akkor M' elutasít, és ha M elfogadó állapotban áll meg, akkor M' elfogad. Tehát $L(M') = \overline{L}$.

Az L_d-t eldöntő TG-et a következőképpen konstruálhatjuk meg: w input esetén először eldöntjük, hogy ez egy TG kódja-e. Ha nem, akkor elutasítjuk. Ha igen, akkor futtatjuk az M' TG-et a $w\#w$ inputon. Ha ez elfogad, akkor $w \notin L(M_w)$, azaz $w \in L_d$, tehát elfogadjuk. Ha M' elutasít, akkor $w \in L(M_w)$, azaz $w \notin L_d$, tehát elutasítjuk.

9. Legyen $\Sigma = \{\mathtt{0}, \mathtt{1}, +\}$. Vázoljon egy Turing-gépet, amelyik az $x{+}y$ alakú inputon (ahol $x, y \in \{\mathtt{0}, \mathtt{1}\}^*$ nem üres bitsorozatok) egy idő után megáll, és az 5. szalagon az x és y binárisan felírt számok összege áll. Adjon becslést a Turing-gép lépésszámára!

Megoldás: Használjunk 5 szalagot. Először a bemenetről a $+$-ig terjedő részt (x) a szokott módon lemásoljuk a 2. szalagra (előtte megjelölve a szalag elejét). Majd a következő részt (y) hasonlóan átmásoljuk a 3. szalagra. Álljunk vissza a fejjel a 2. és 3. szalag utolsó nem üres mezőjére (azaz x és y utolsó bitjére), hiszen az összeadást az utolsó bitnél jó kezdeni.

A 4. szalagra írjuk az összeget a szám végéről kezdve. Az összeadásnál két állapotot használunk, az egyik jelképezi, hogy volt átvitel az előző számjegy összeadásából, a másik meg, hogy nem volt. Ennek megfelelően a két állapotban mást-mást írunk a 4. szalagra ugyanannál az olvasott karakterpárnál. Végül a 4. szalagon fordított sorrendben levő eredményt másoljuk a jó sorrendben az 5. szalagra.

A lépésszám $O(n)$.

10. Oldjuk meg az előző feladatot összeadás helyett szorzás esetére is.

    Megoldás: Az iskolában tanult algoritmust kell megvalósítani. A lépésszám $O(n^2)$.