

Számítógépes Hálózatok

8. gyakorlat

Elérhetőségek

- honlap: <http://szalaigj.web.elte.hu/>
- email: szalaigindl@inf.elte.hu
- szoba: 2.507 (déli tömb)

Gyakorlat tematika

- Csúszó ablak protokoll
- Minimális keretméret
- Bináris visszaszámlálás protokoll
- Adaptív fa protokoll
- Órai / házi feladat

Csúszó ablak – emlékeztető

- Egy adott időpontban egyszerre több keret is átviteli állapotban lehet.
- A keret sorozatbeli pozíciója adja a keret címkéjét (sorozatszám); a küldő nyilvántartja a küldhető sorozatszámok halmazát (adási ablak), a fogadó pedig a fogadhatóakét (vételi ablak)
- Mi van ha egy hosszú folyam közepén történik egy keret hiba?

Csúszó ablak – emlékeztető

- „visszalépés N-nel” (Go-Back-N):
 - az összes hibás keret utáni keretet eldobja és nyugtát sem küld róluk;
 - mikor az adónak lejár az időzítője, akkor újraküldi az összes nyugtázatlan keretet, kezdve a sérült vagy elveszett kerete
 - <https://www.youtube.com/watch?v=9BuaeEjleQI>
- „szelektív ismétlés” (Selective Repeat):
 - a hibás kereteket eldobja, de a jó kereteket a hibás után pufferelemi;
 - mikor az adónak lejár az időzítője, akkor a legrégebbi nyugtázatlan keretet küldi el újra
 - <https://www.youtube.com/watch?v=Cs8tR8A9jm8>

Gyakorló feladat 1.

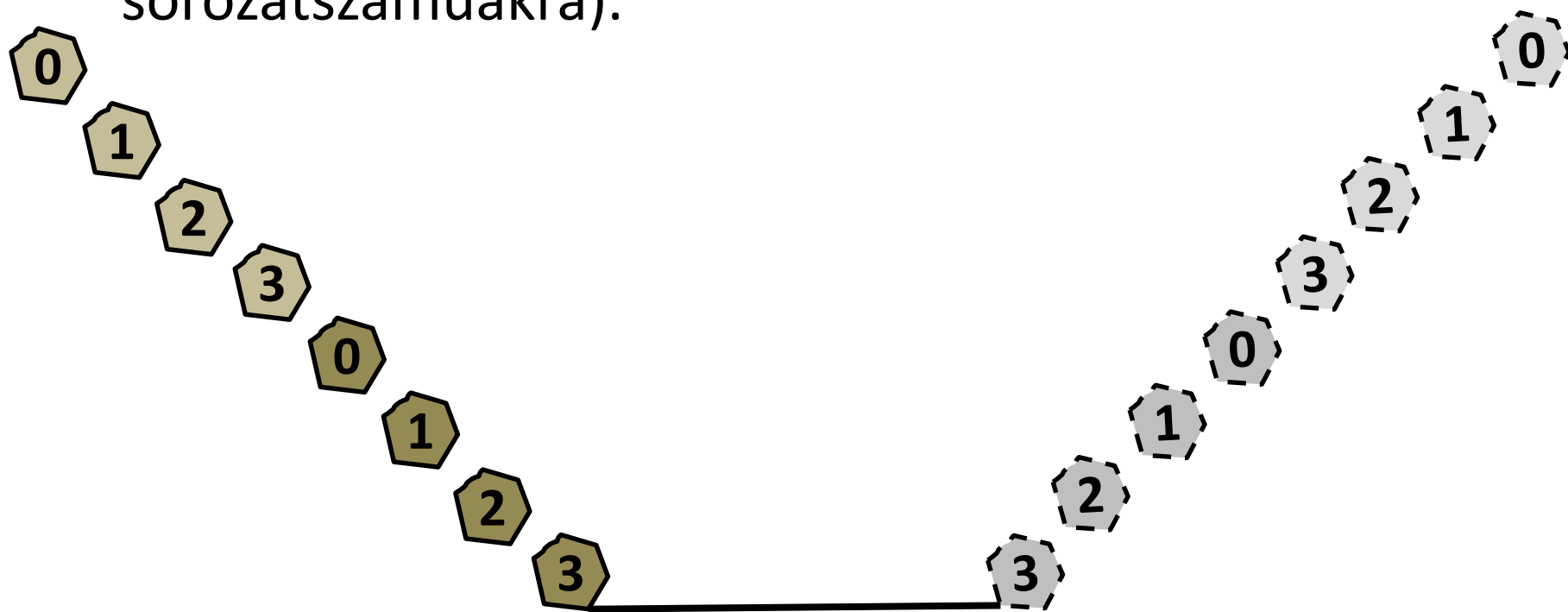
- A Go-Back-N és Selective Repeat esetén legfeljebb hány csomagot küldhet a küldő egyszerre, illetve legfeljebb hány csomag lehet egyidejűleg elküldött, de nem nyugtázott, ha a sorszámok tere 4 elemű (pl. sorszámok 0-tól 3-ig)?
- Gondoljunk a legkedvezőtlenebb pillanatokban elveszett nyugtákra.
- Mutassunk egy példát erre az esetre.

Gyakorló feladat 1. megoldása

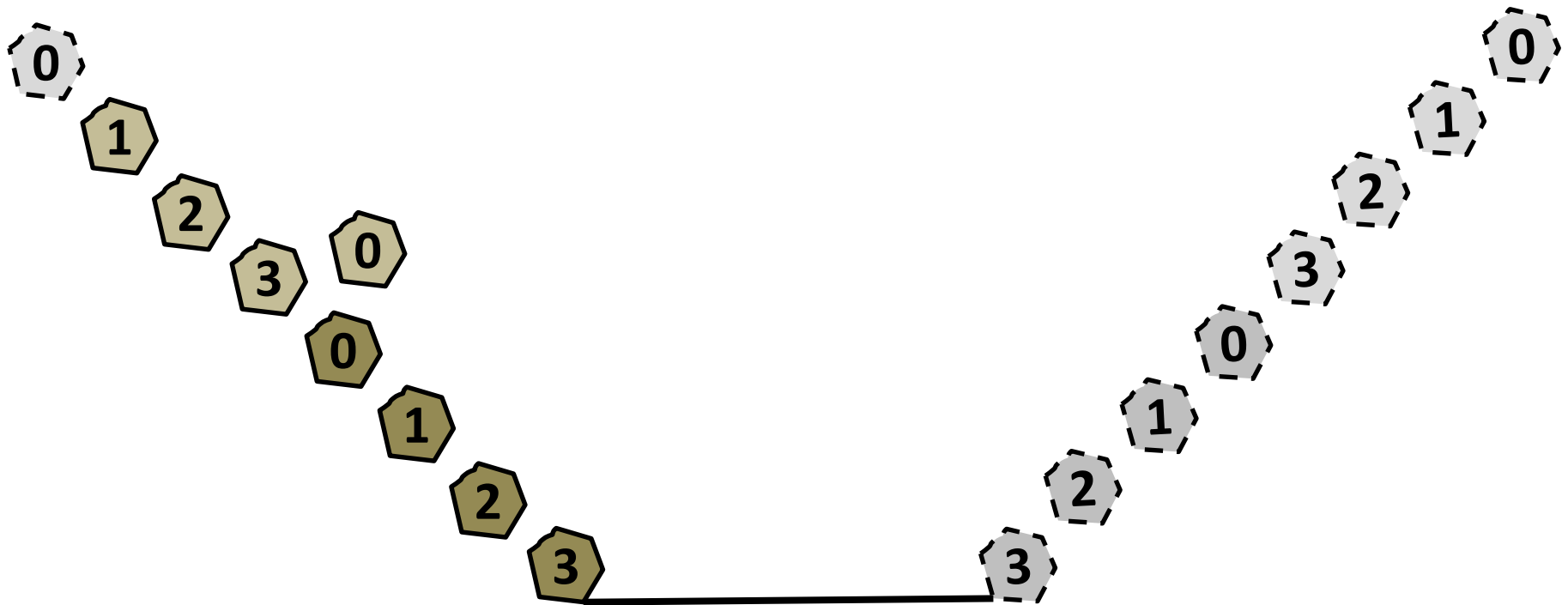
- A lényeg mindkét esetben, hogy a fogadónak tudnia kell megkülönböztetni egy már fogadott keret újraküldését egy új keret kezdeti/eredeti küldésétől.
- Go-Back-N: ekkor maximum 3 lehet egyidejűleg elküldött.
- (Mindkét szimuláció a http://webmuseum.mi.fh-offenburg.de/exhibition/73/assets/sliding_win.swf alapján készült)

Gyakorló feladat 1. megoldása

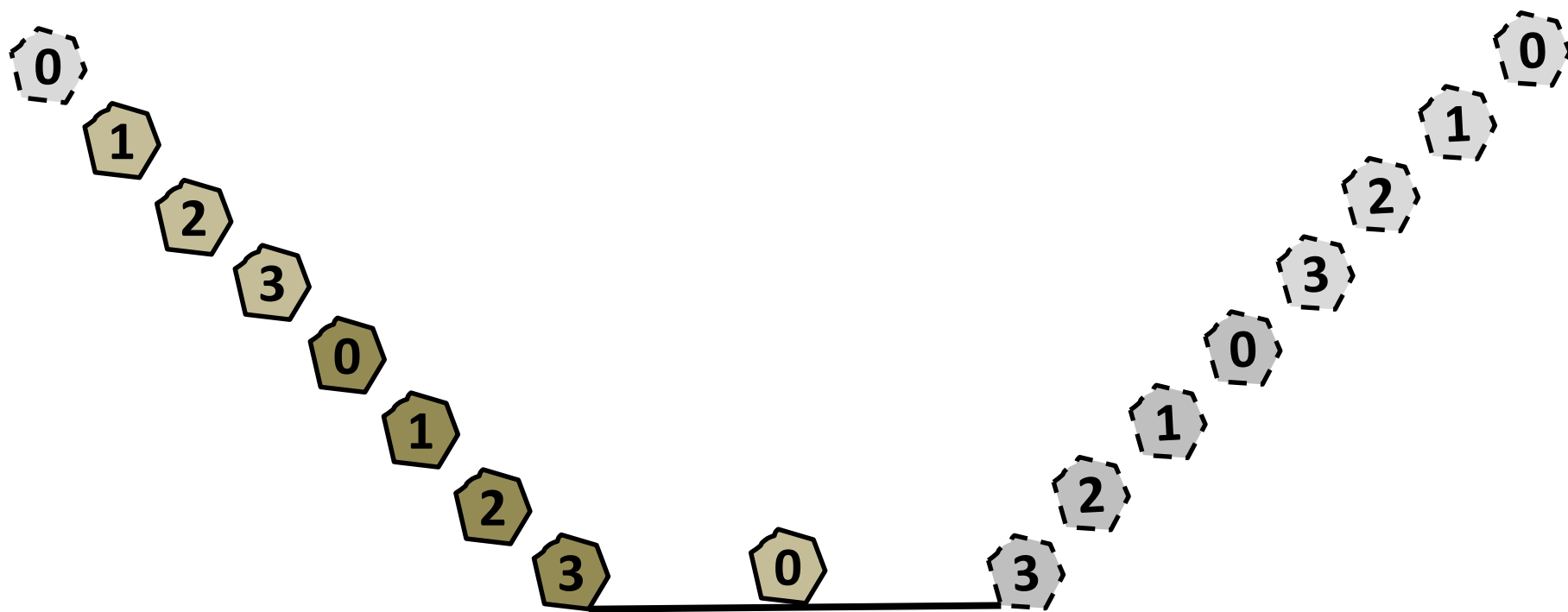
- Mivel ha 4-re lenne beállítva, akkor a fogadó várakozna az első négy keretre (0...3 sorozatszámúakra).



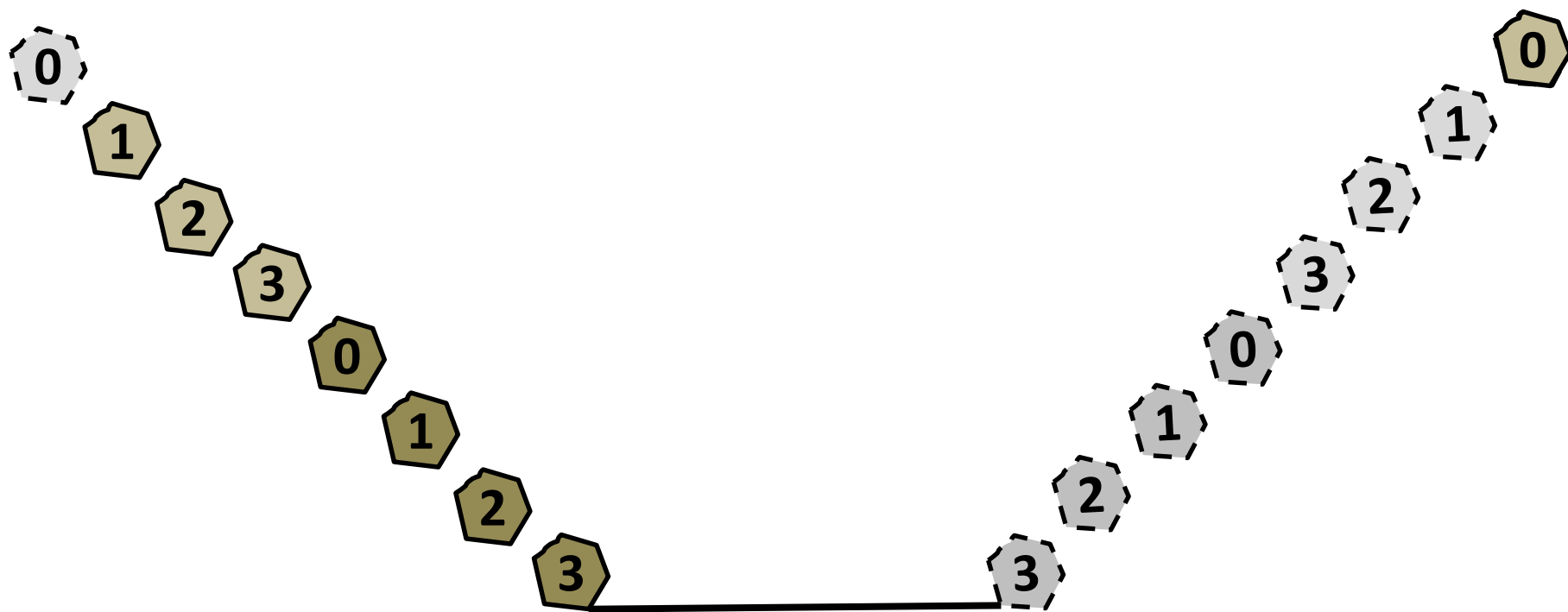
Gyakorló feladat 1. megoldása



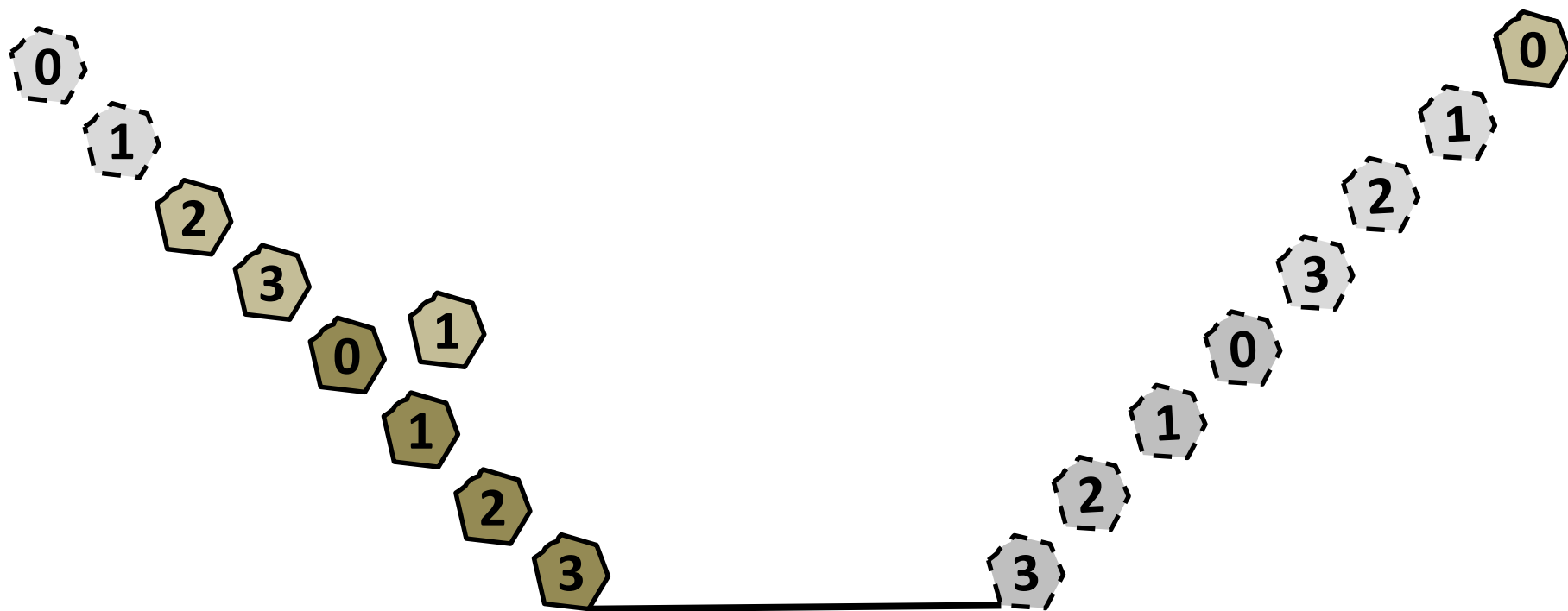
Gyakorló feladat 1. megoldása



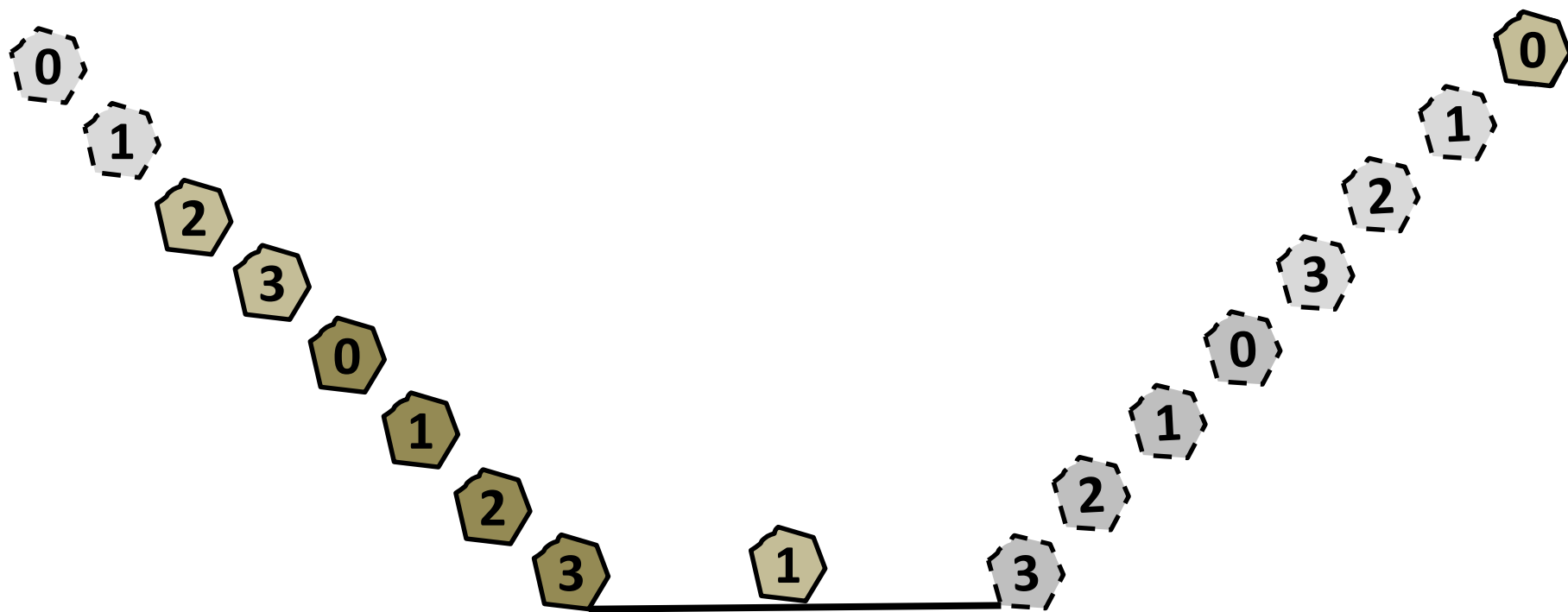
Gyakorló feladat 1. megoldása



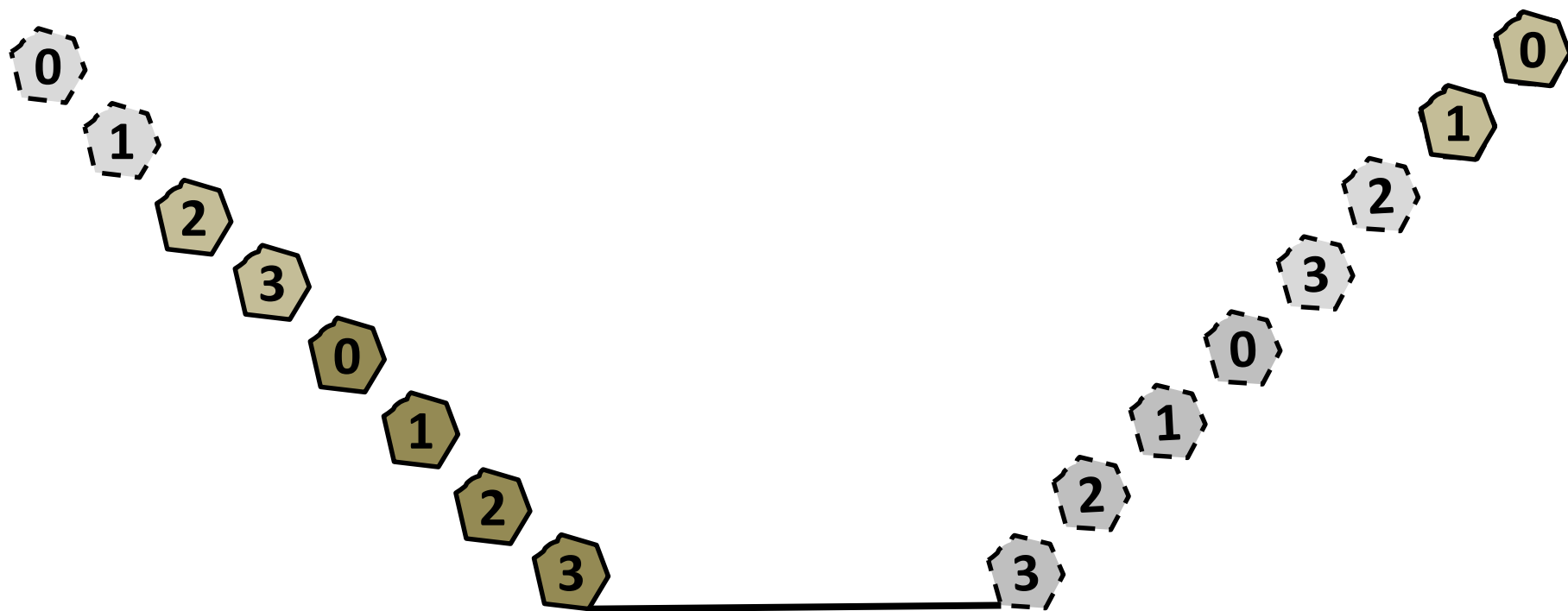
Gyakorló feladat 1. megoldása



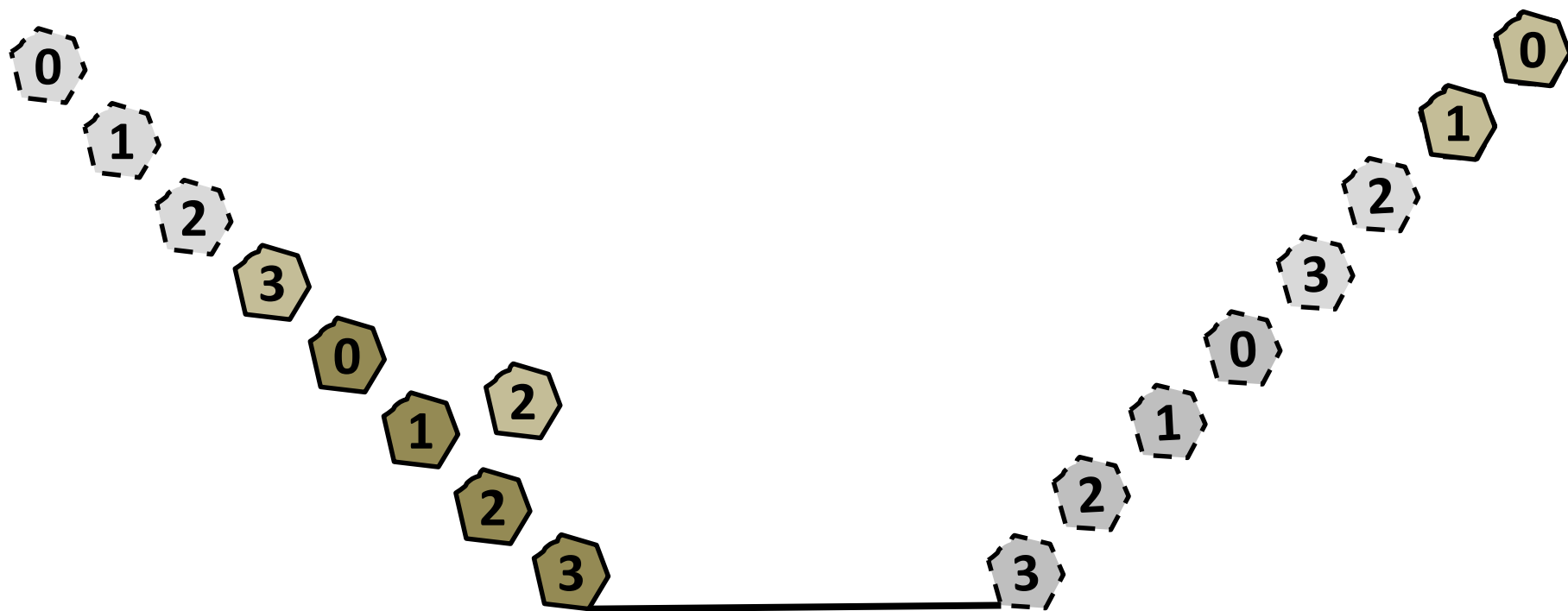
Gyakorló feladat 1. megoldása



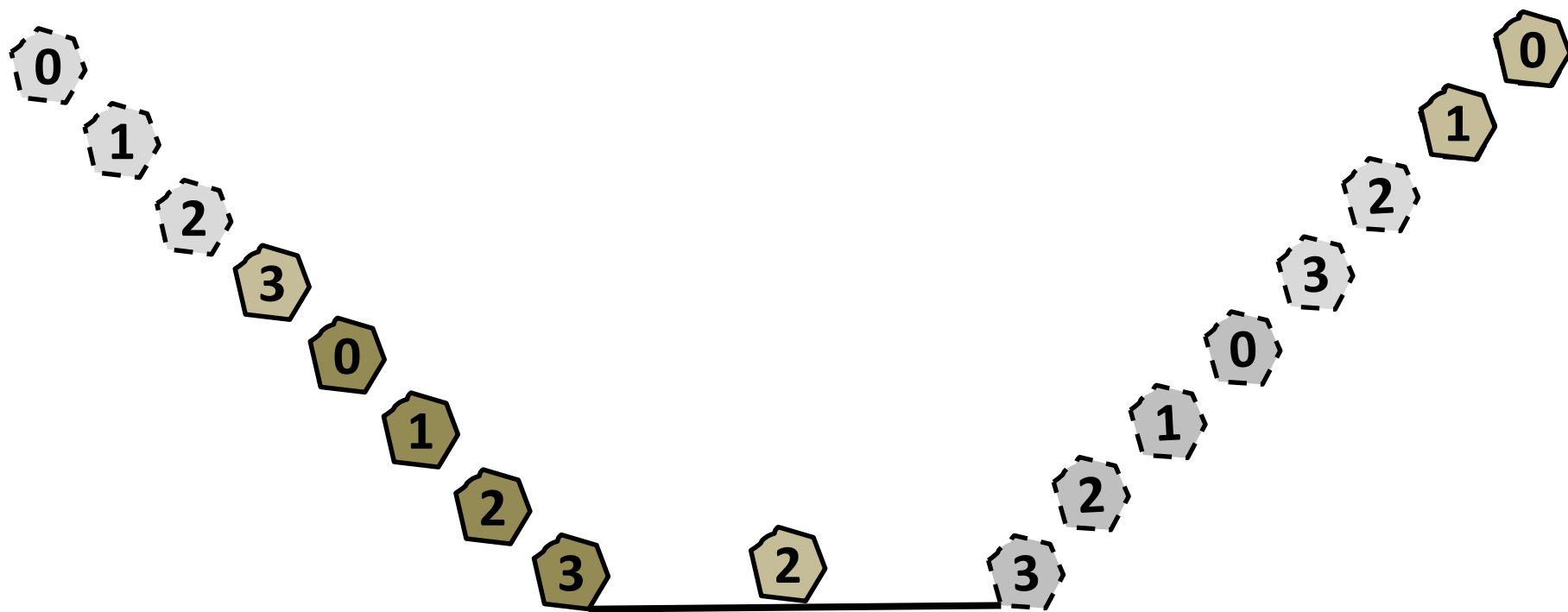
Gyakorló feladat 1. megoldása



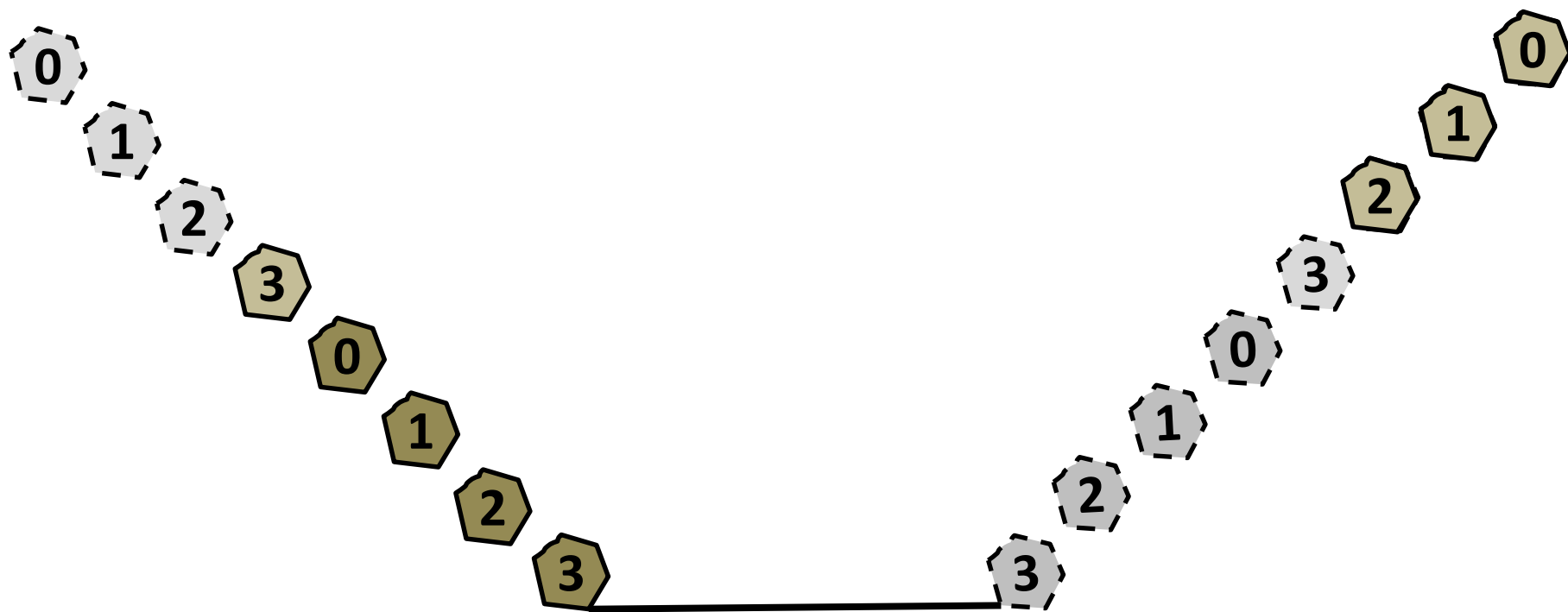
Gyakorló feladat 1. megoldása



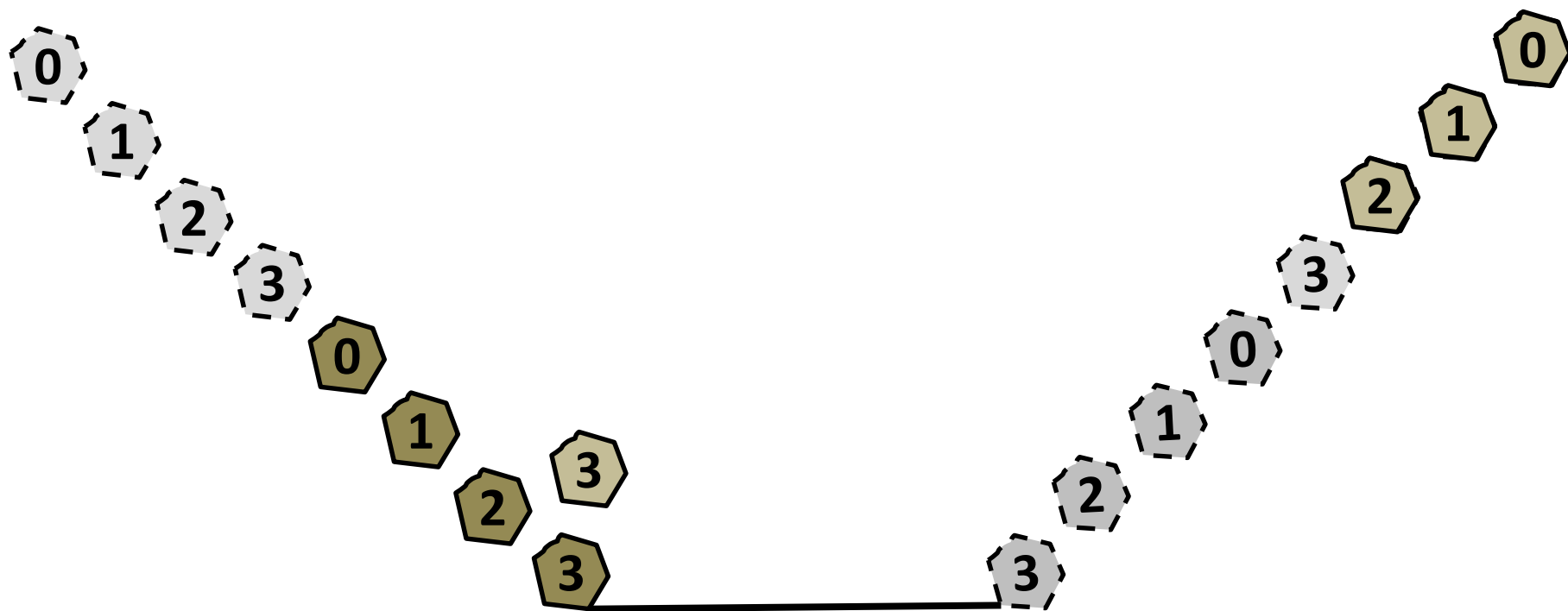
Gyakorló feladat 1. megoldása



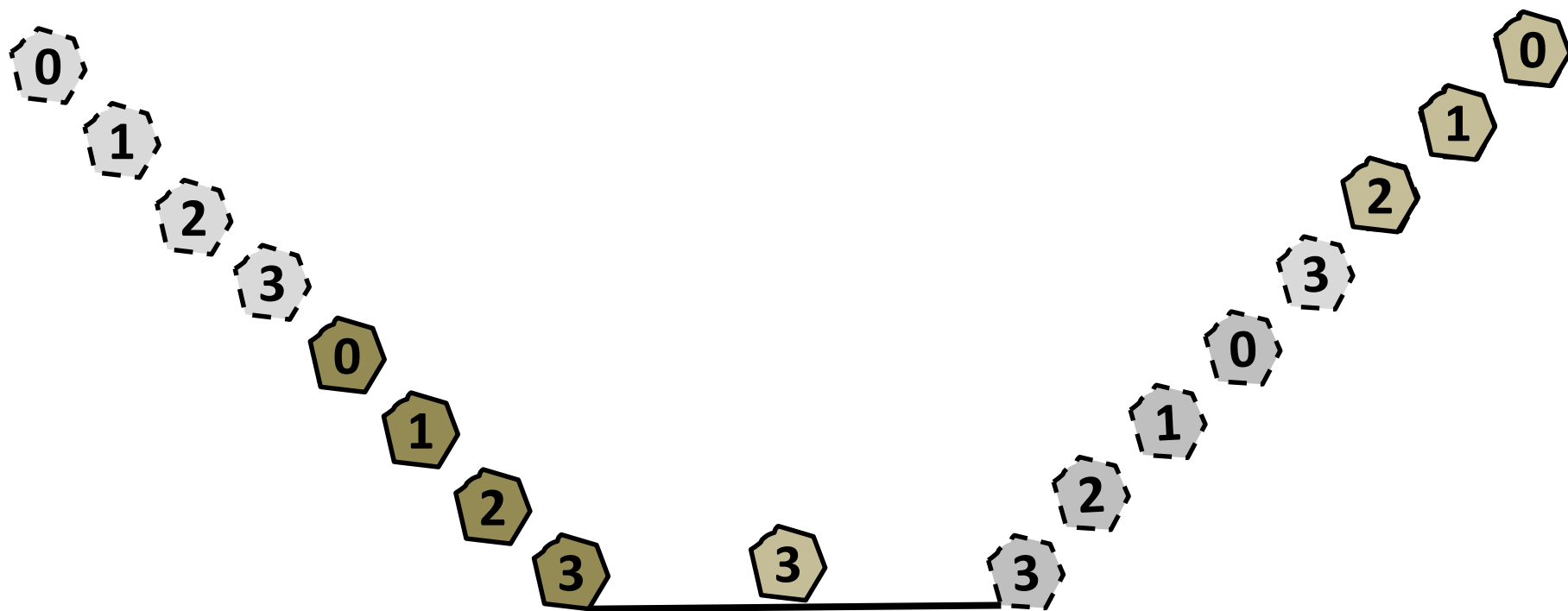
Gyakorló feladat 1. megoldása



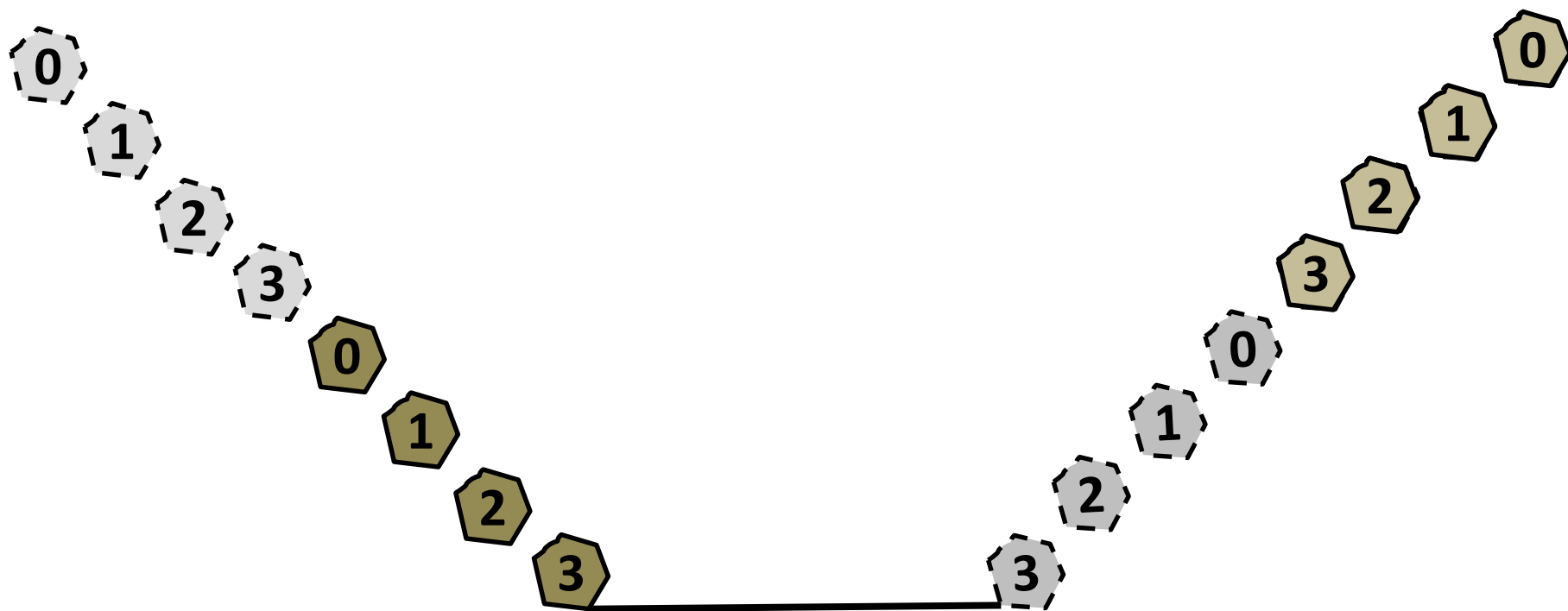
Gyakorló feladat 1. megoldása



Gyakorló feladat 1. megoldása

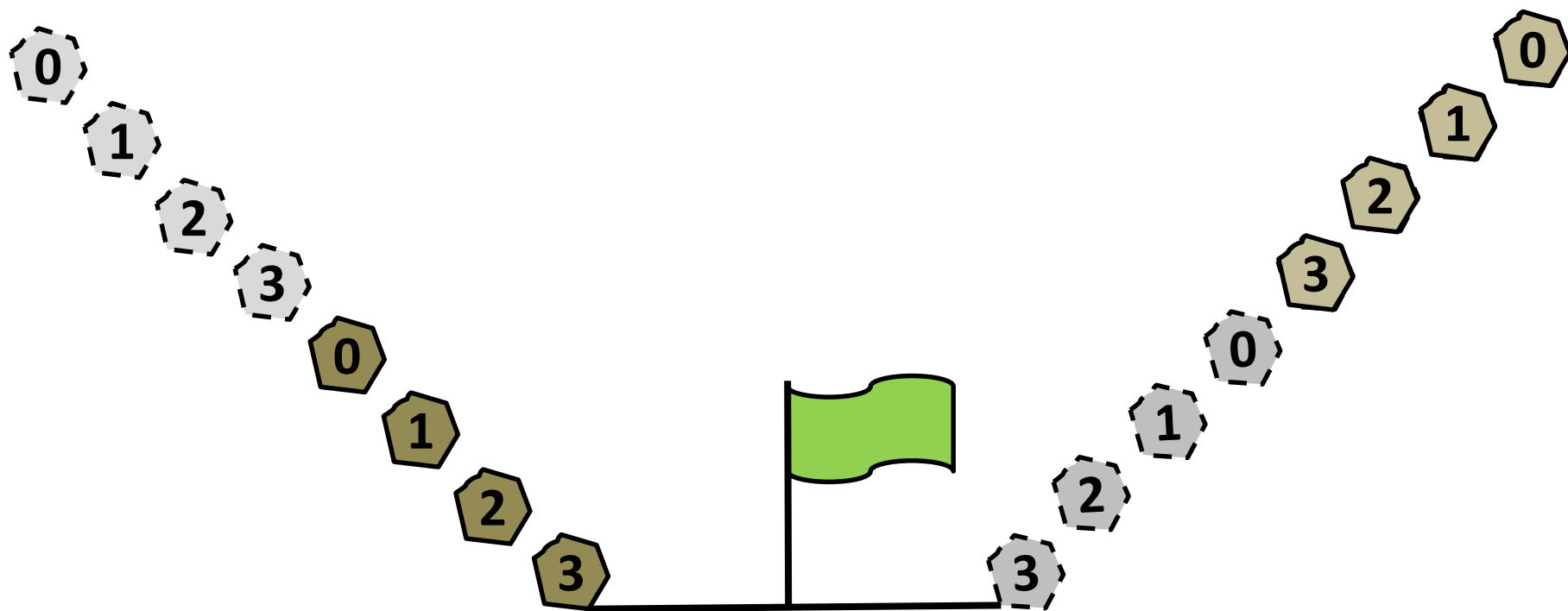


Gyakorló feladat 1. megoldása



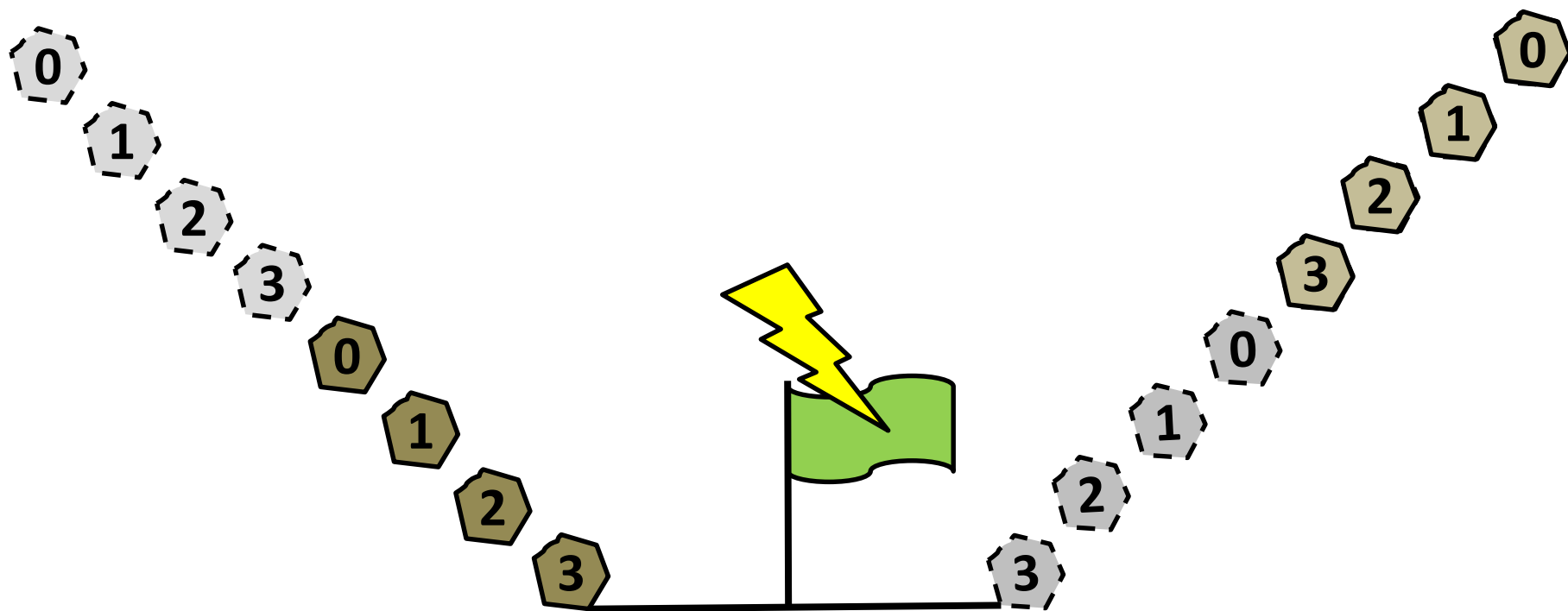
Gyakorló feladat 1. megoldása

- Ha megérkeznének, akkor küldené vissza a nyugtákat.



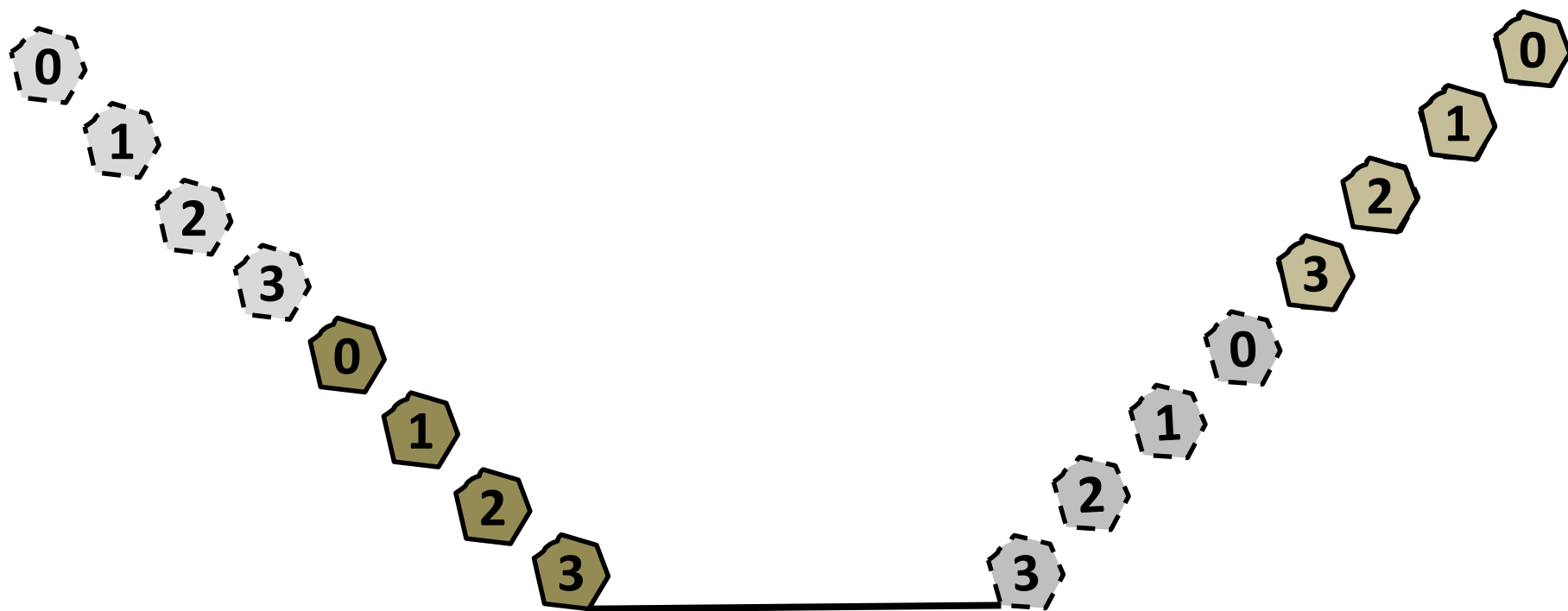
Gyakorló feladat 1. megoldása

- De tegyük fel, hogy azok elvesznének.



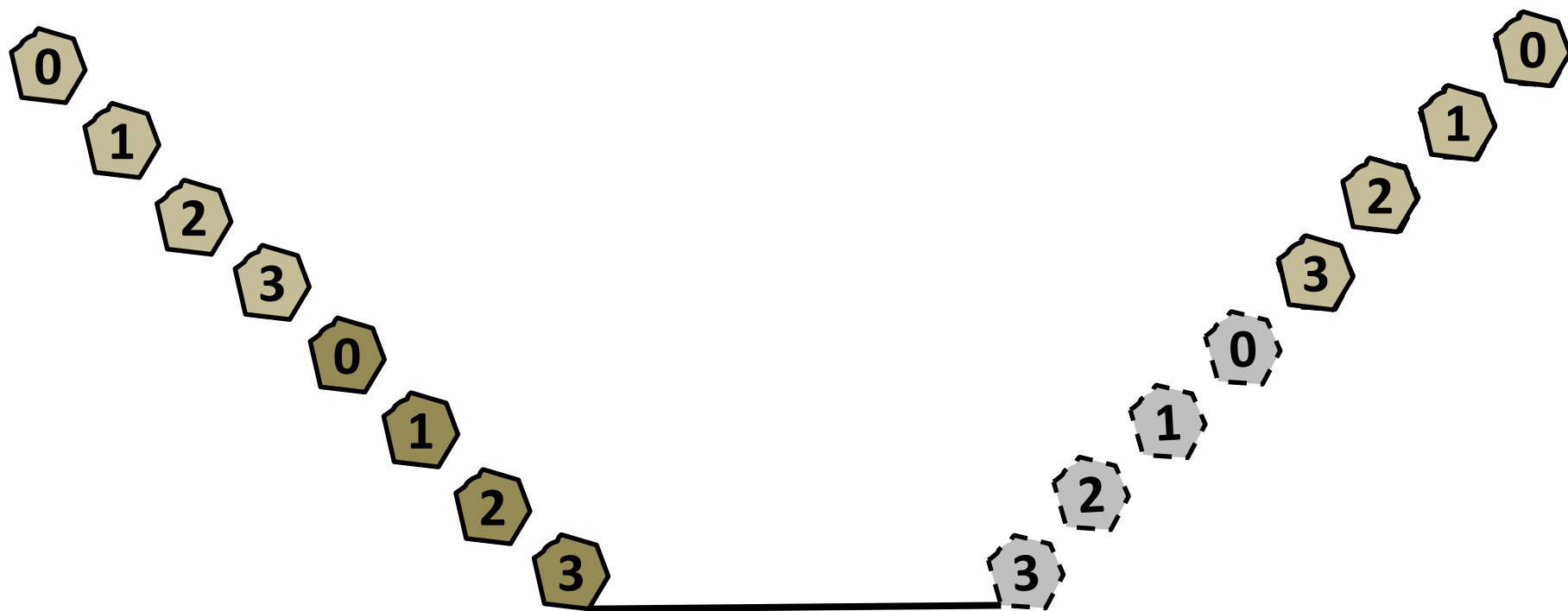
Gyakorló feladat 1. megoldása

- A küldő a timeout után...

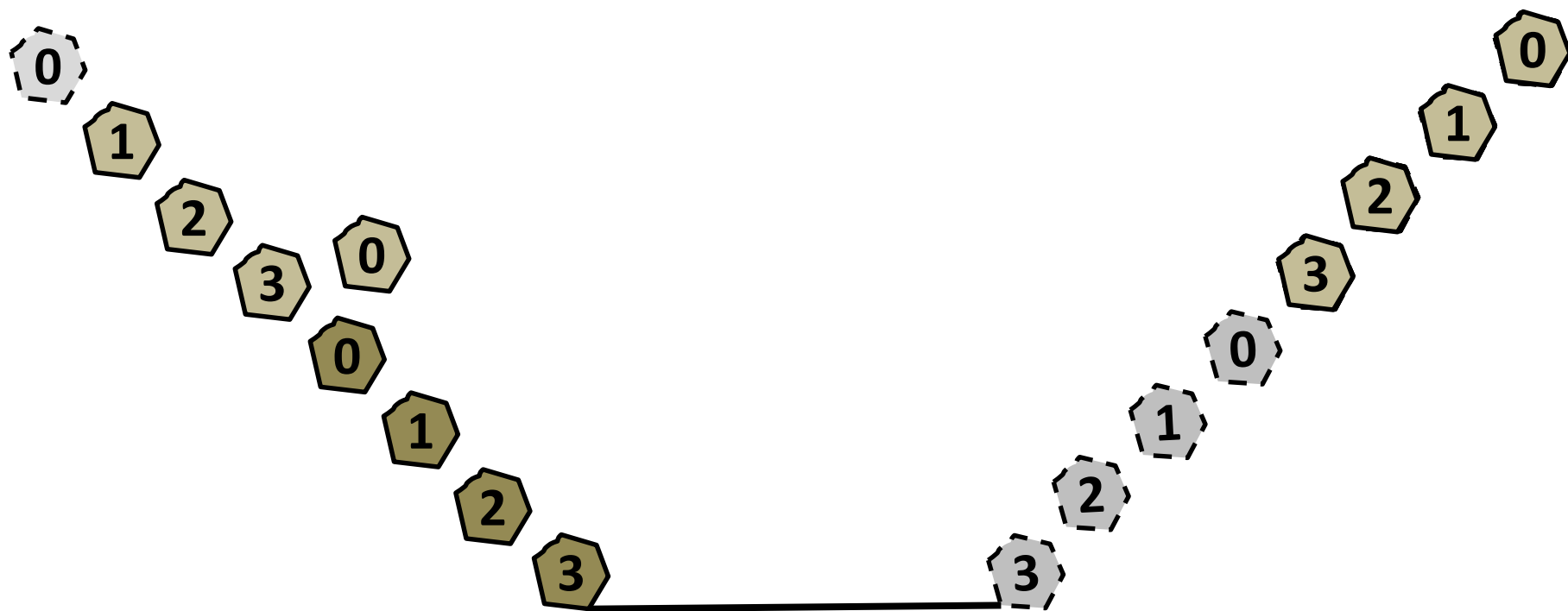


Gyakorló feladat 1. megoldása

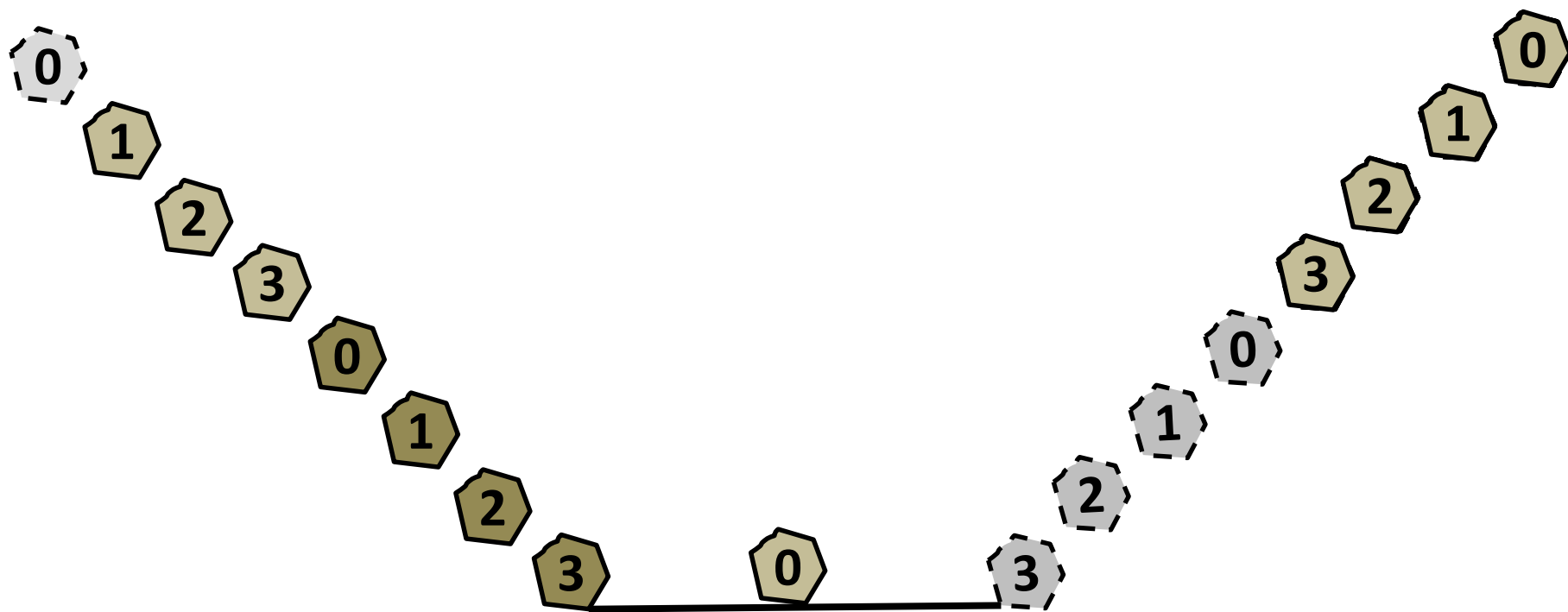
- ... újraküldené az első négy keretet.



Gyakorló feladat 1. megoldása

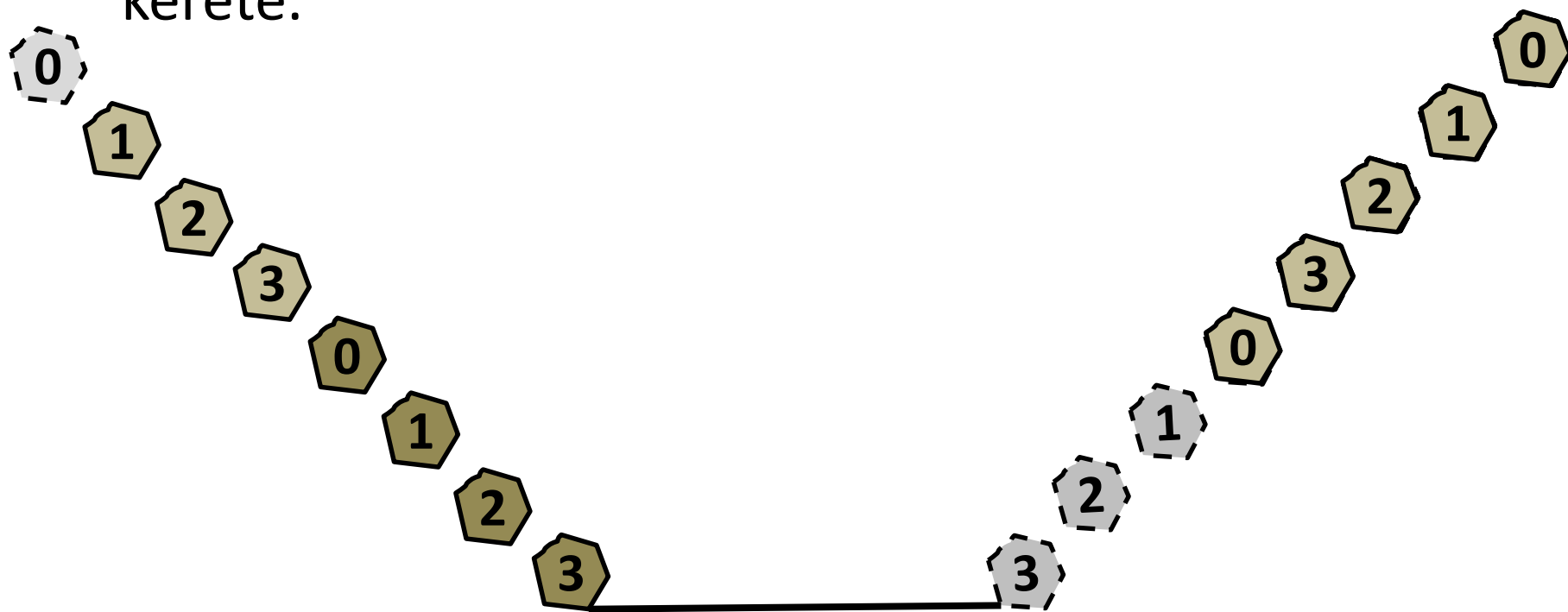


Gyakorló feladat 1. megoldása

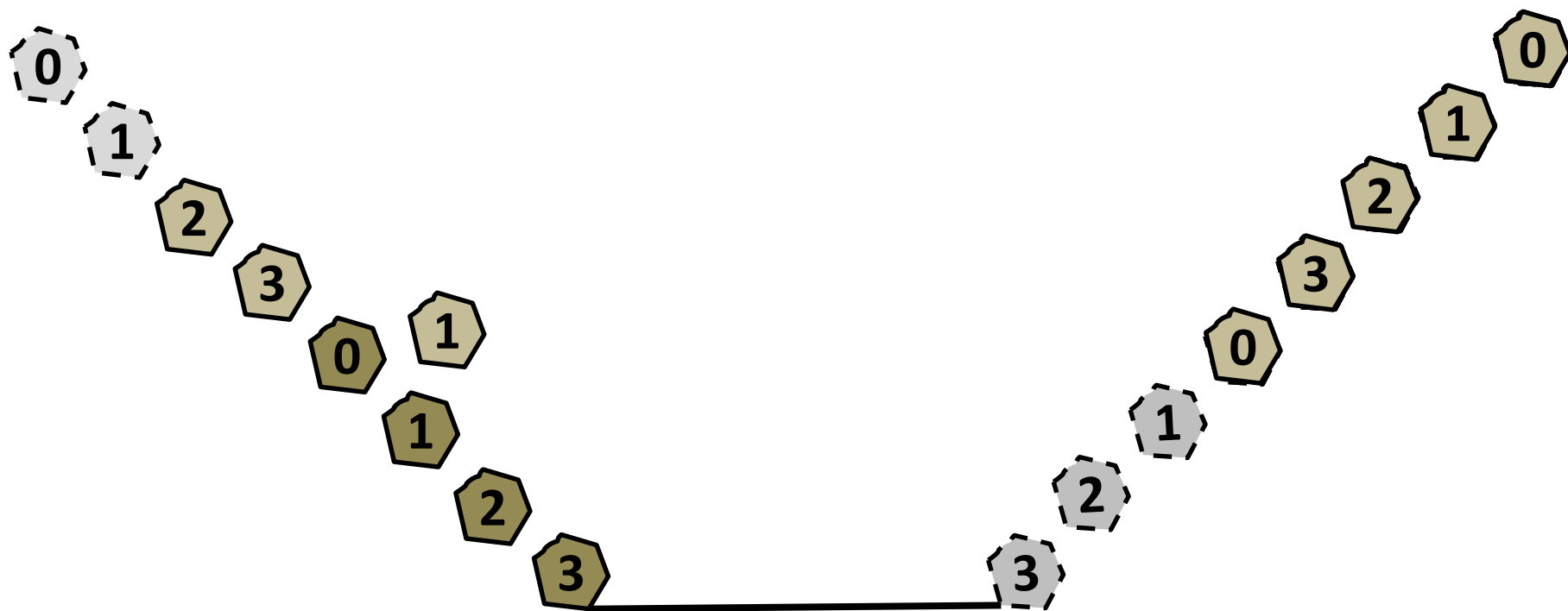


Gyakorló feladat 1. megoldása

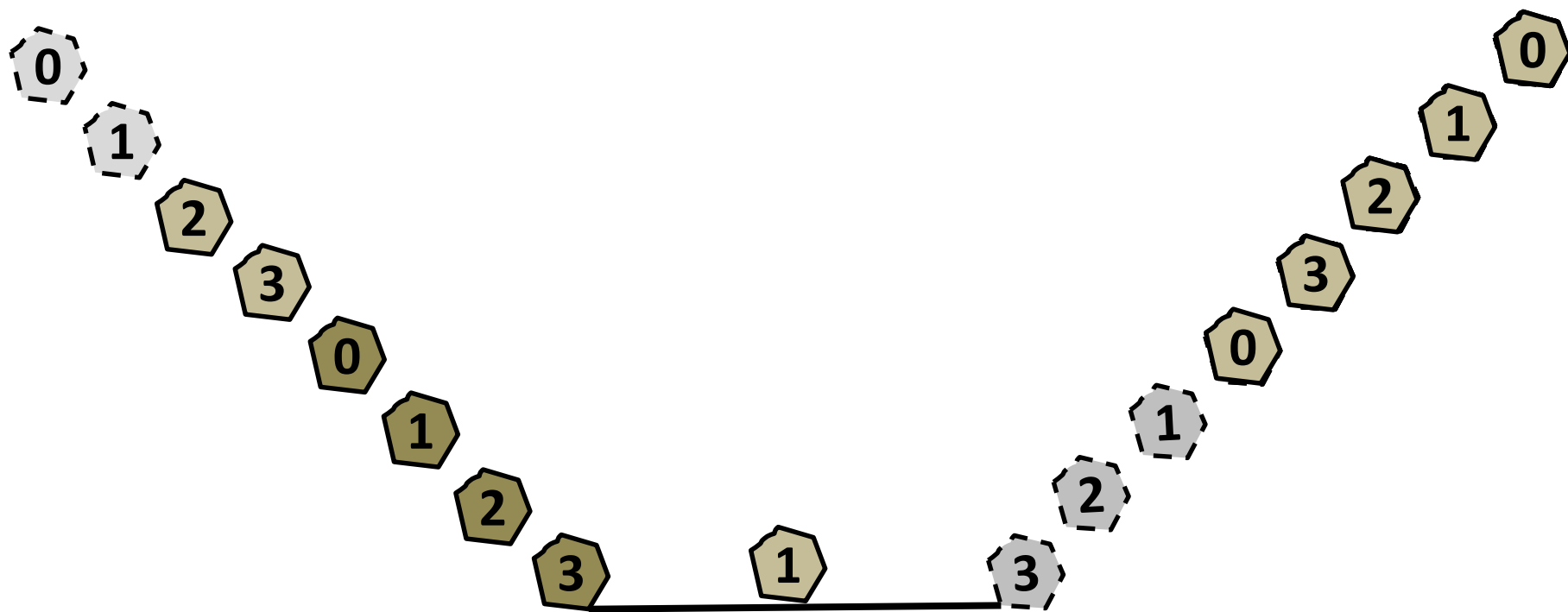
- Amikor a 0 sorozatszámú megérkezik, akkor a fogadó azt hinné, hogy ez már a második sorozat első kerete.



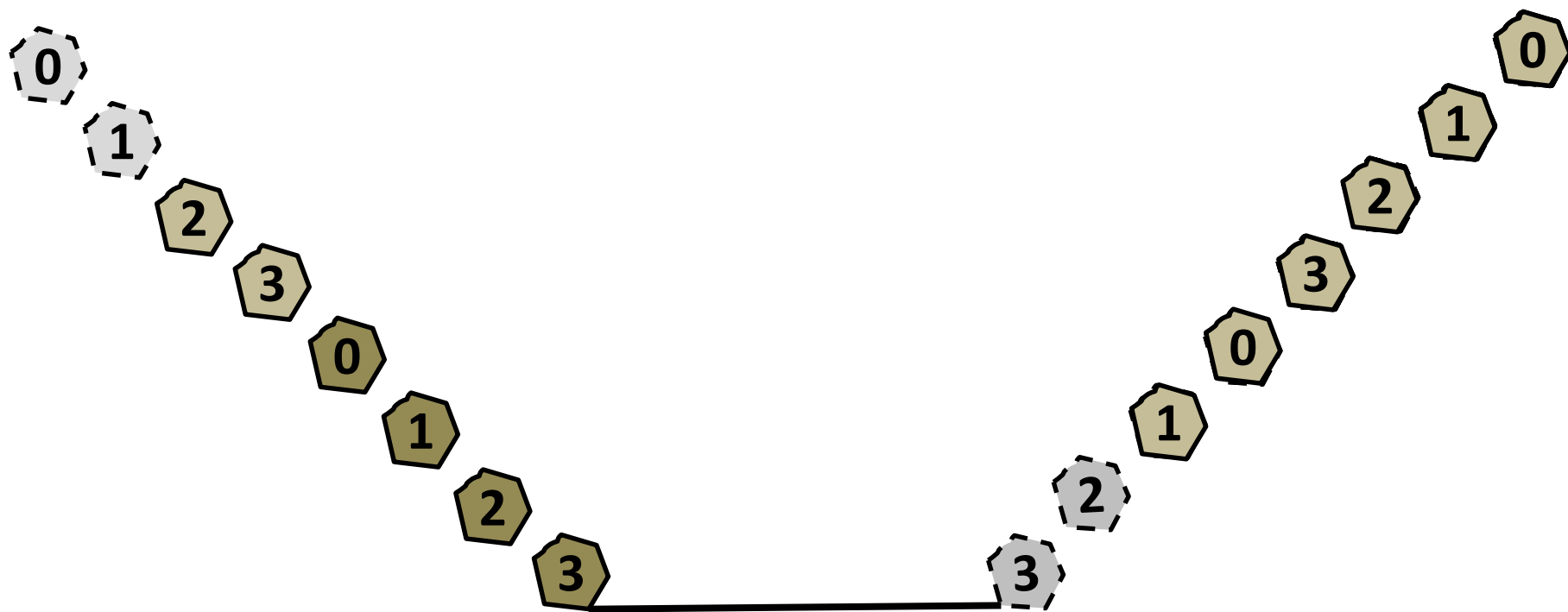
Gyakorló feladat 1. megoldása



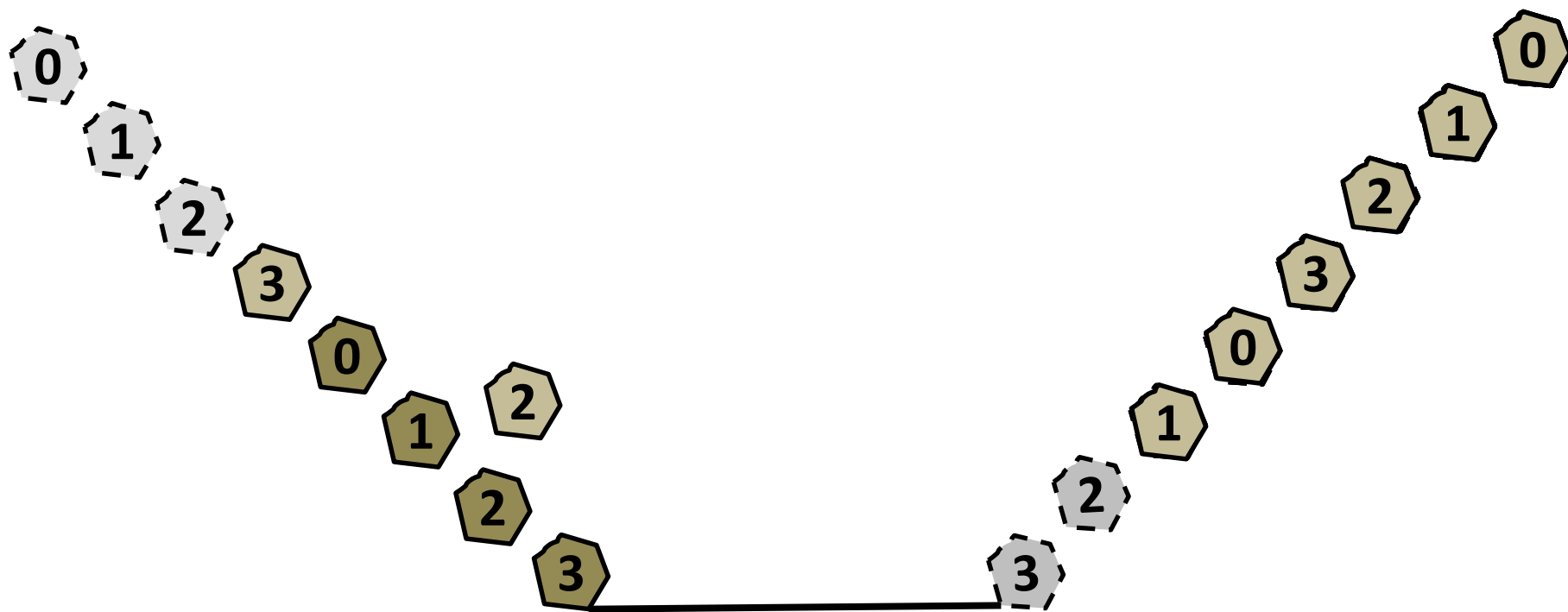
Gyakorló feladat 1. megoldása



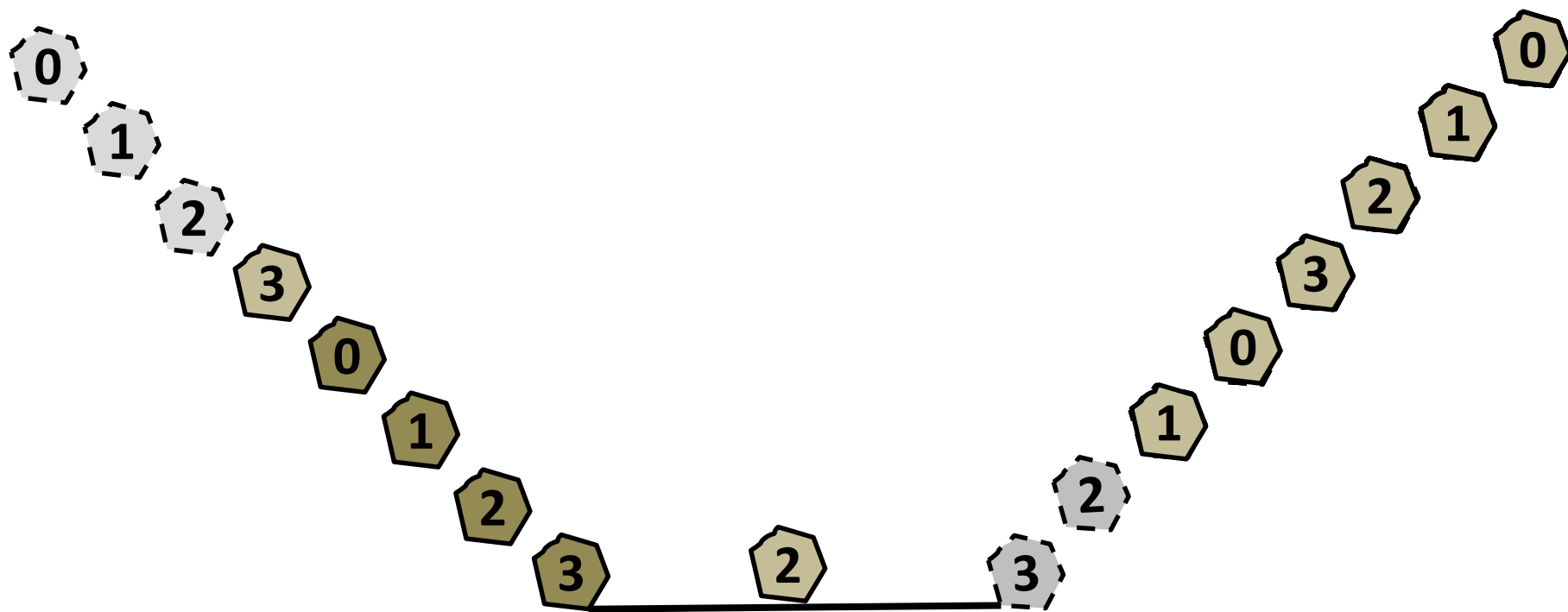
Gyakorló feladat 1. megoldása



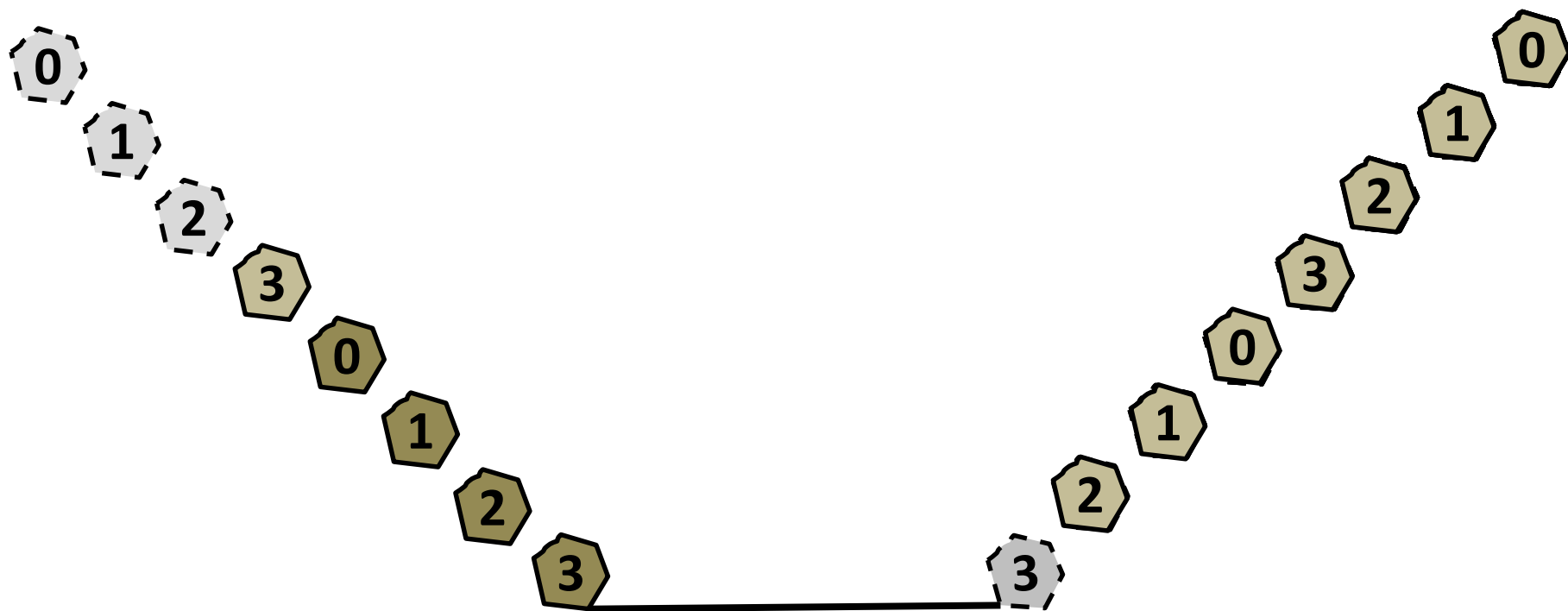
Gyakorló feladat 1. megoldása



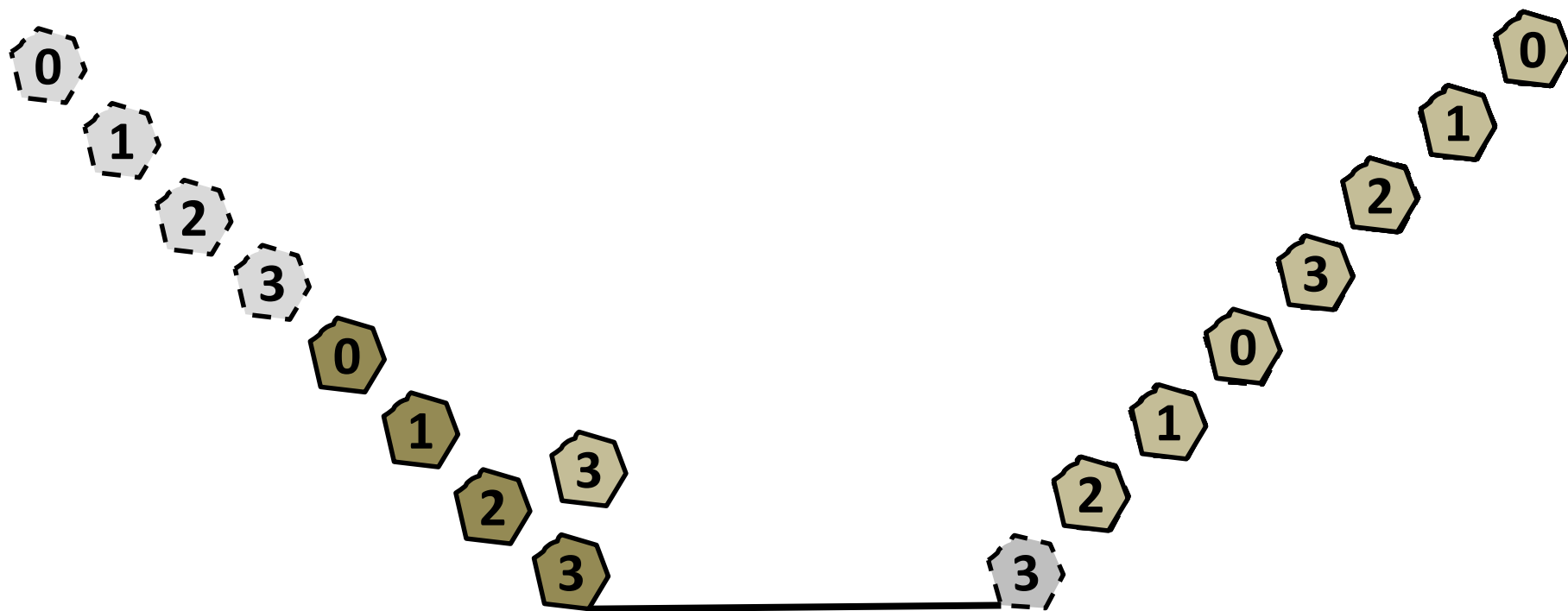
Gyakorló feladat 1. megoldása



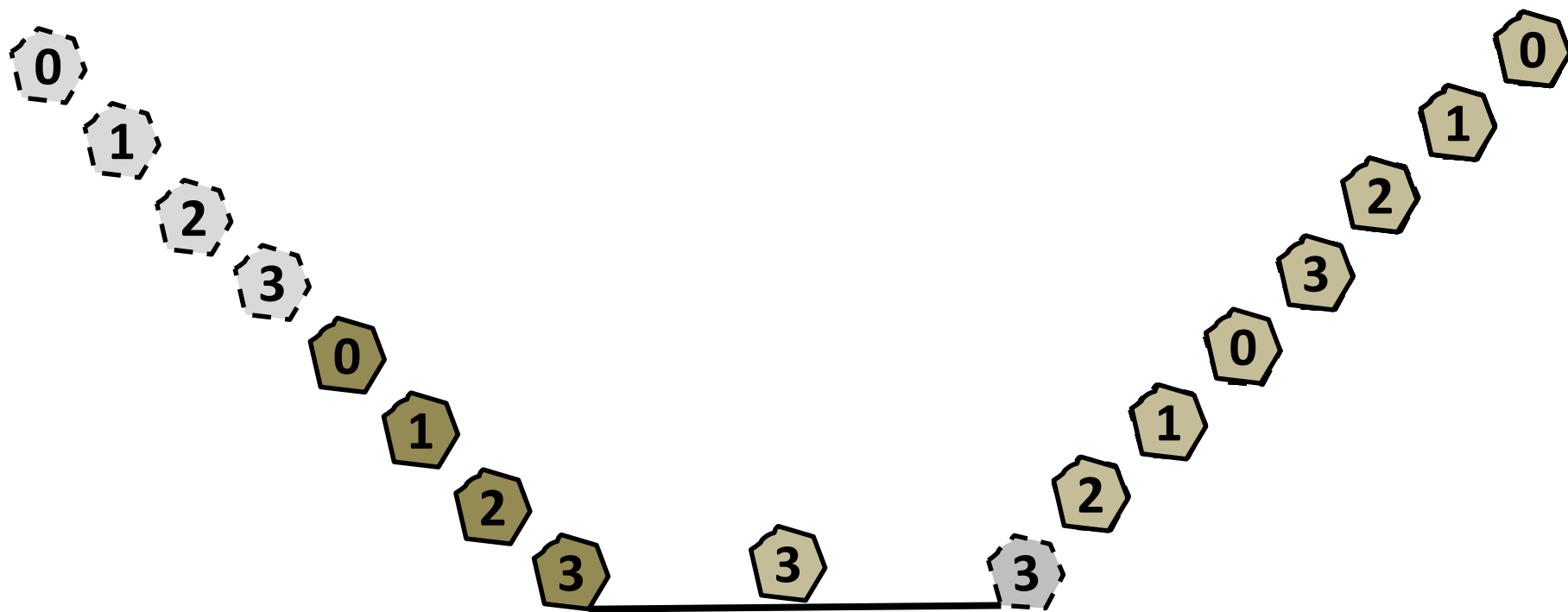
Gyakorló feladat 1. megoldása



Gyakorló feladat 1. megoldása

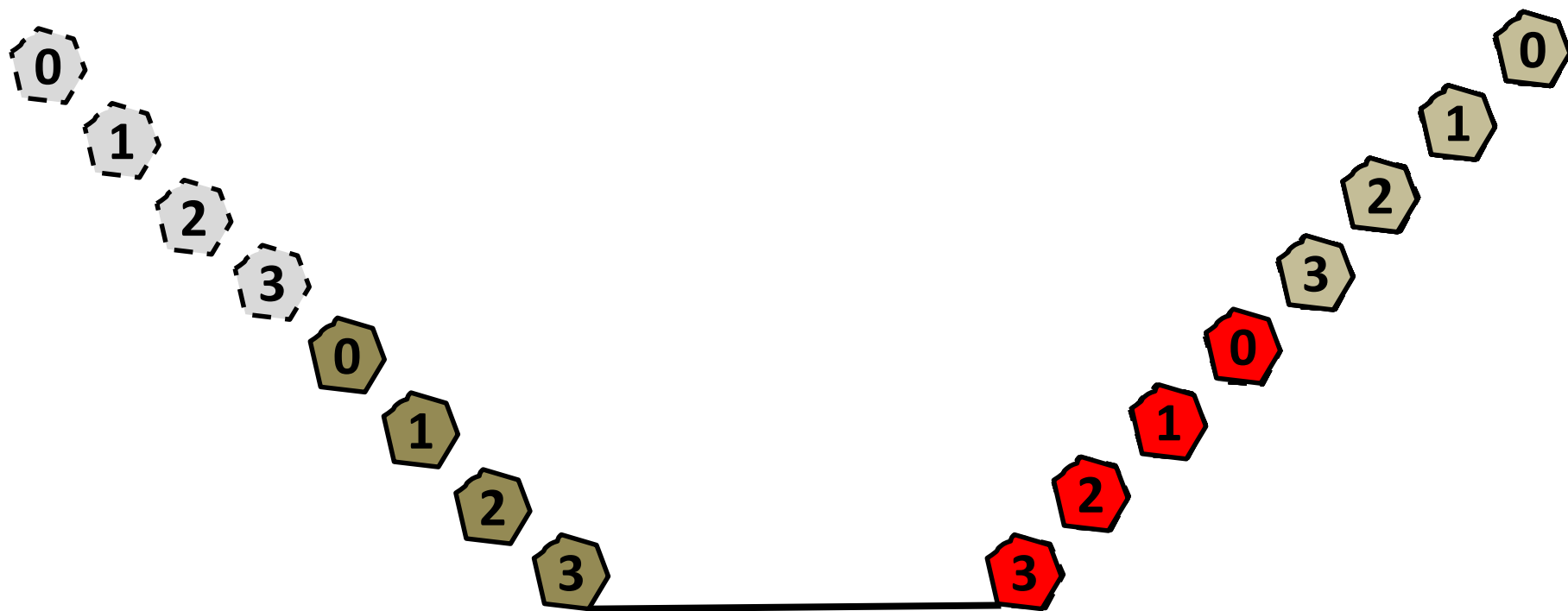


Gyakorló feladat 1. megoldása



Gyakorló feladat 1. megoldása

- Ez hibához vezetne.

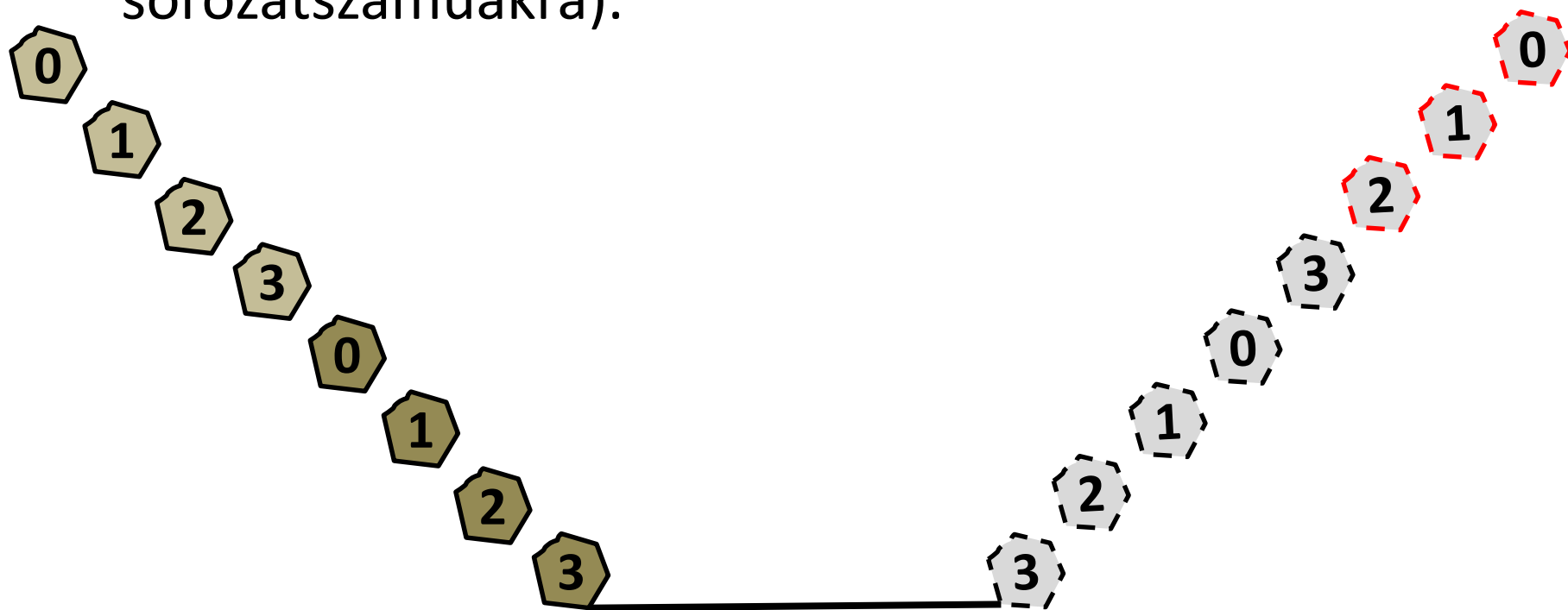


Gyakorló feladat 1. megoldása

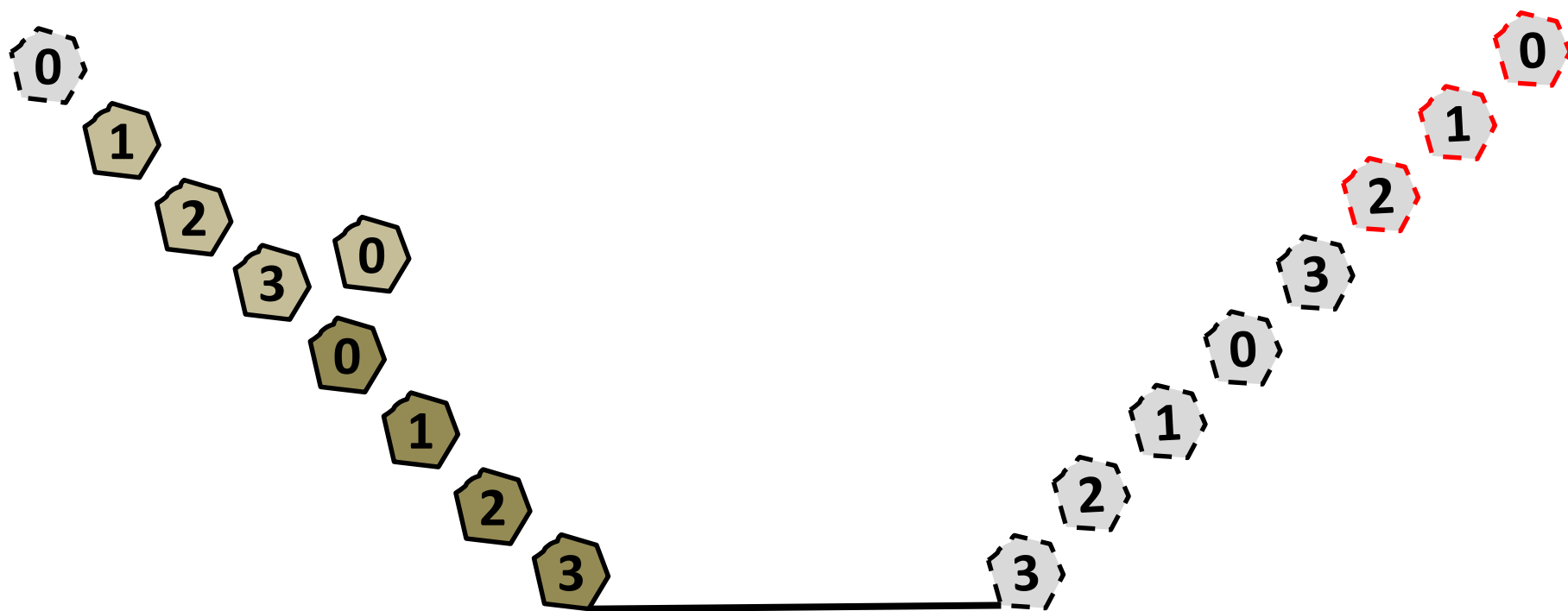
- Selective Repeat: ekkor maximum 2 lehet egyidejűleg elküldött.

Gyakorló feladat 1. megoldása

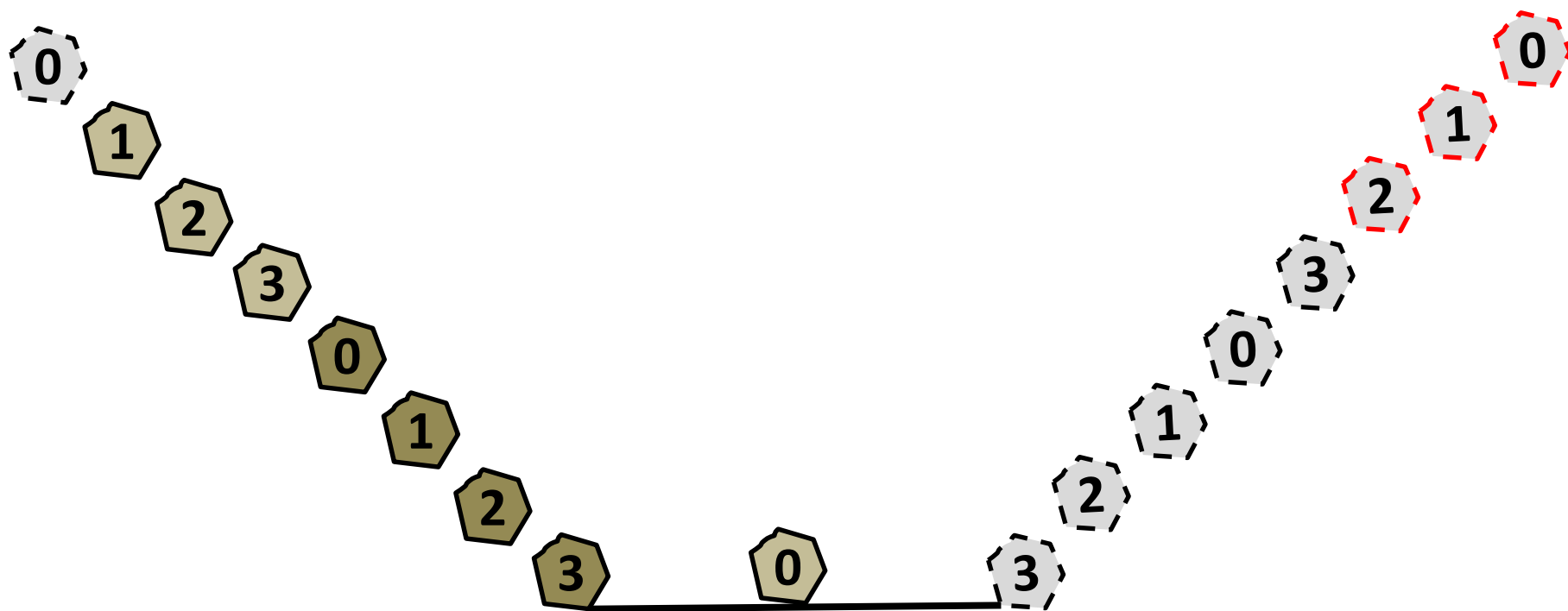
- Mivel ha 3-ra lenne beállítva, akkor a fogadó várakozna az első három keretre (0,1,2 sorozatszámúakra).



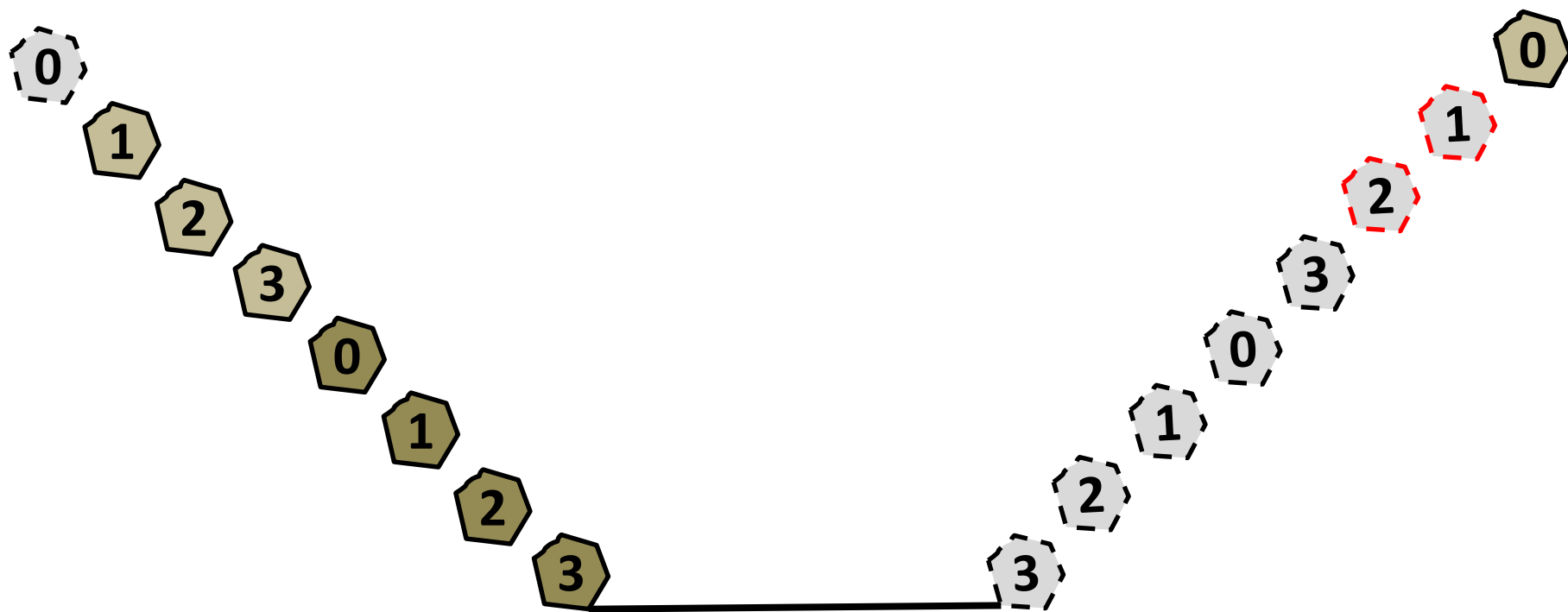
Gyakorló feladat 1. megoldása



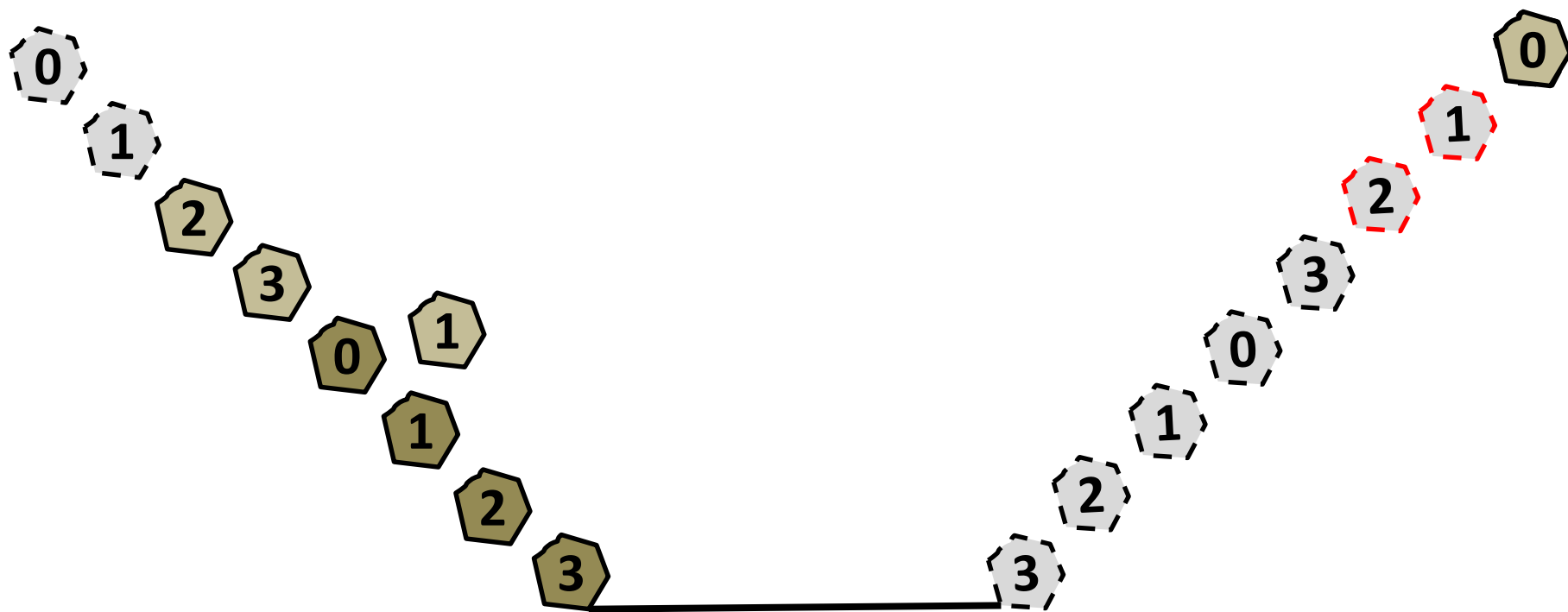
Gyakorló feladat 1. megoldása



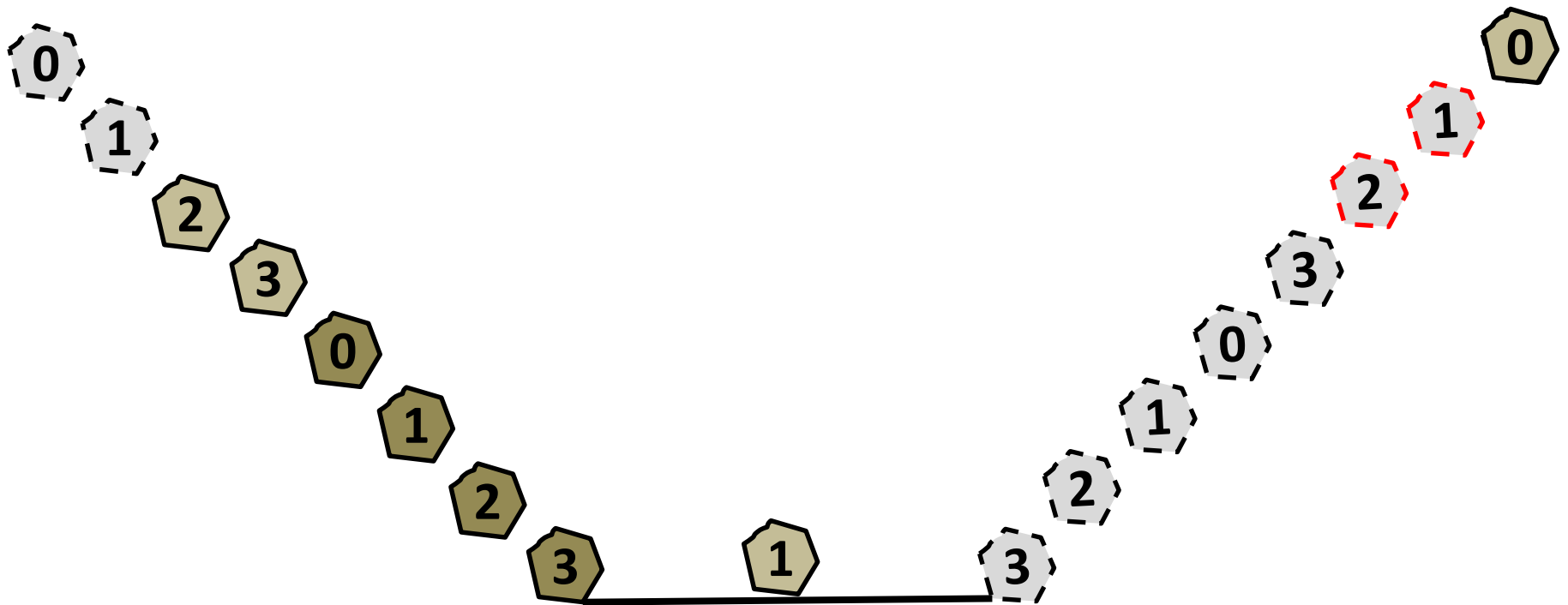
Gyakorló feladat 1. megoldása



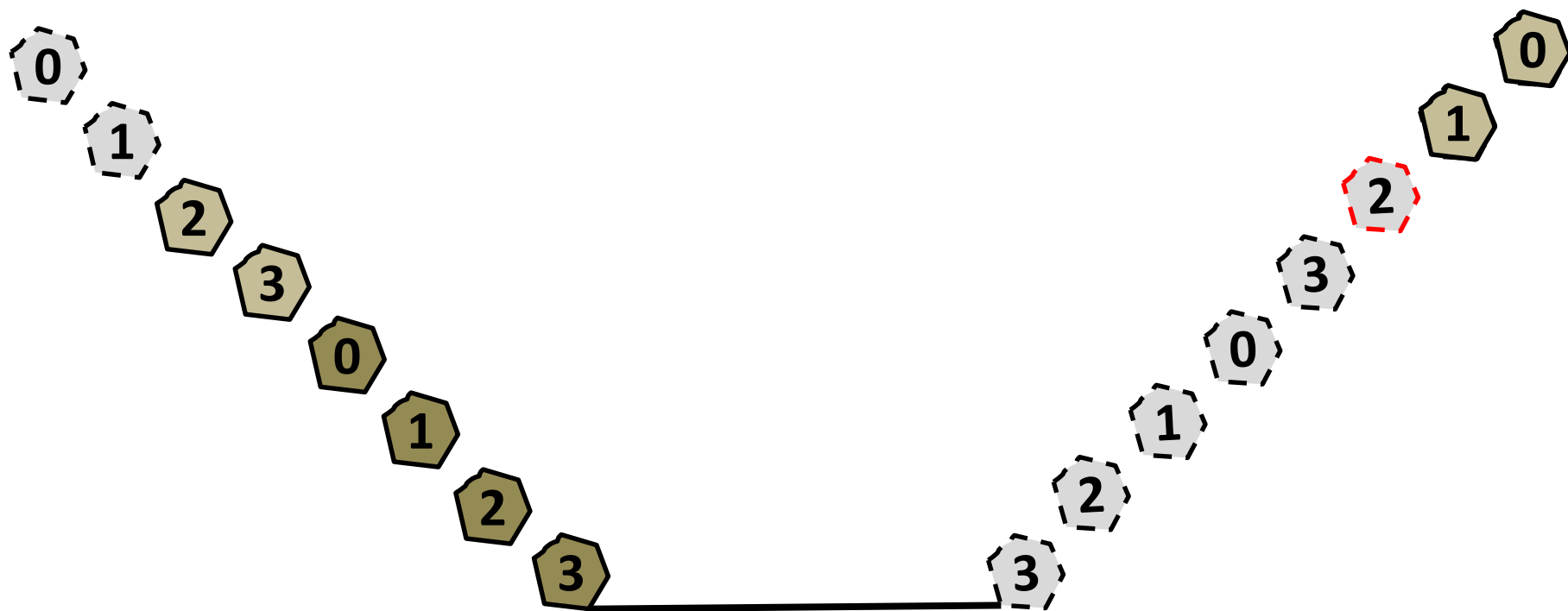
Gyakorló feladat 1. megoldása



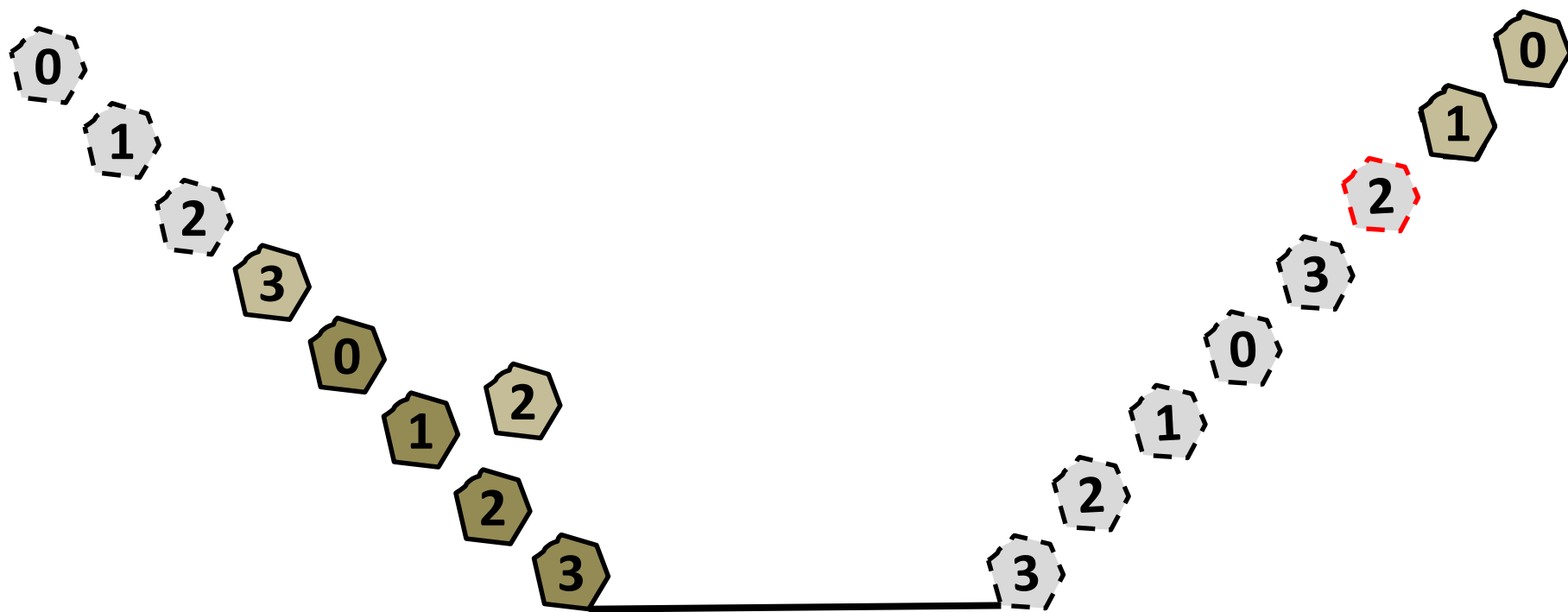
Gyakorló feladat 1. megoldása



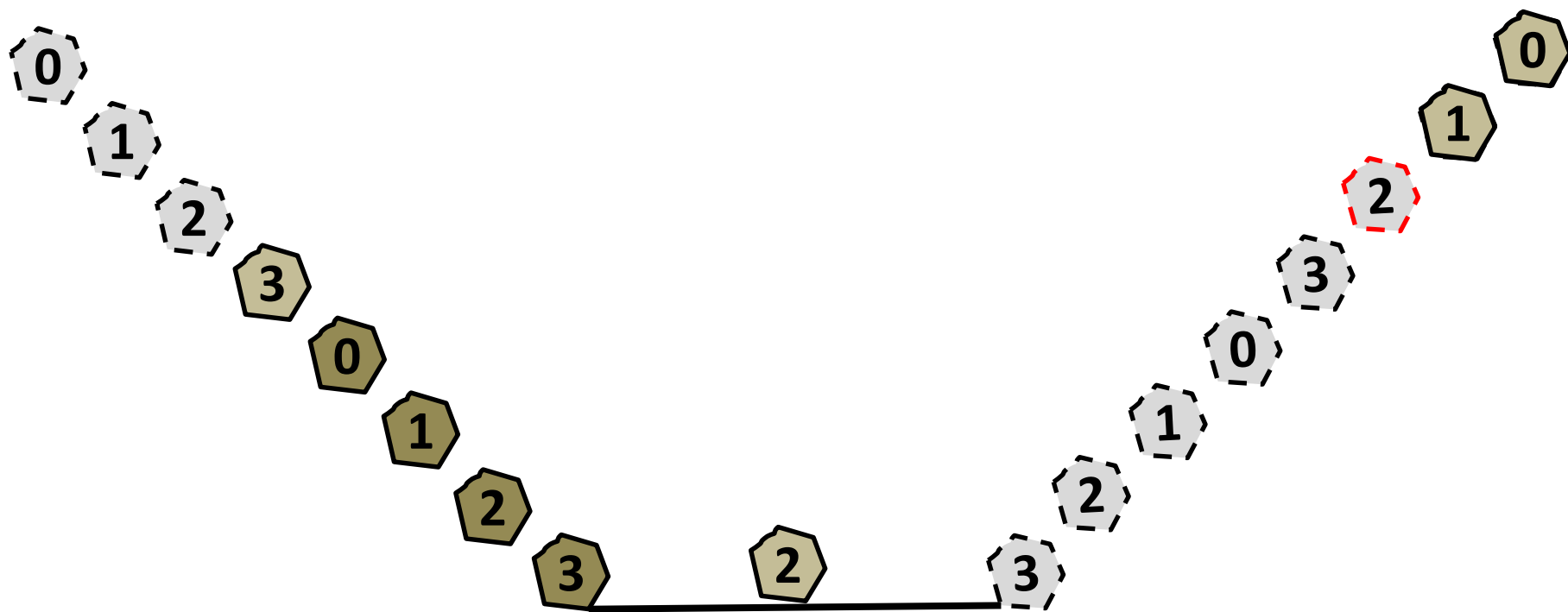
Gyakorló feladat 1. megoldása



Gyakorló feladat 1. megoldása

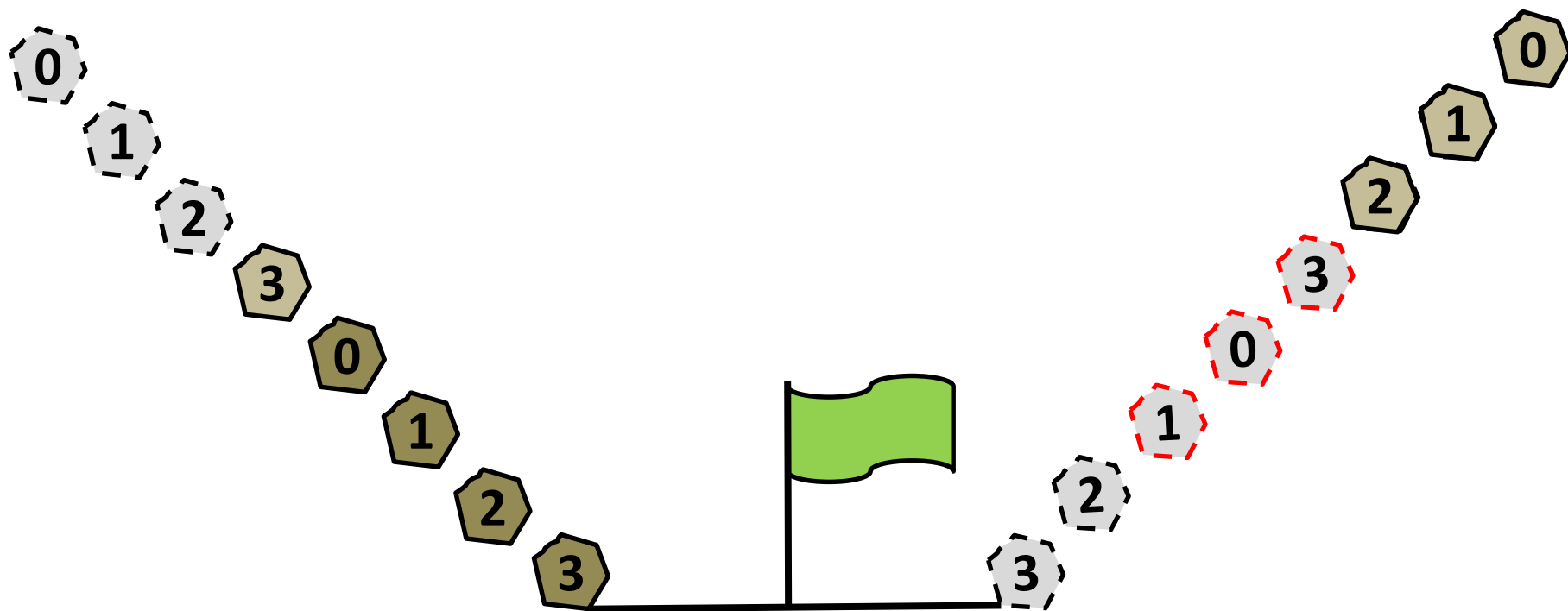


Gyakorló feladat 1. megoldása



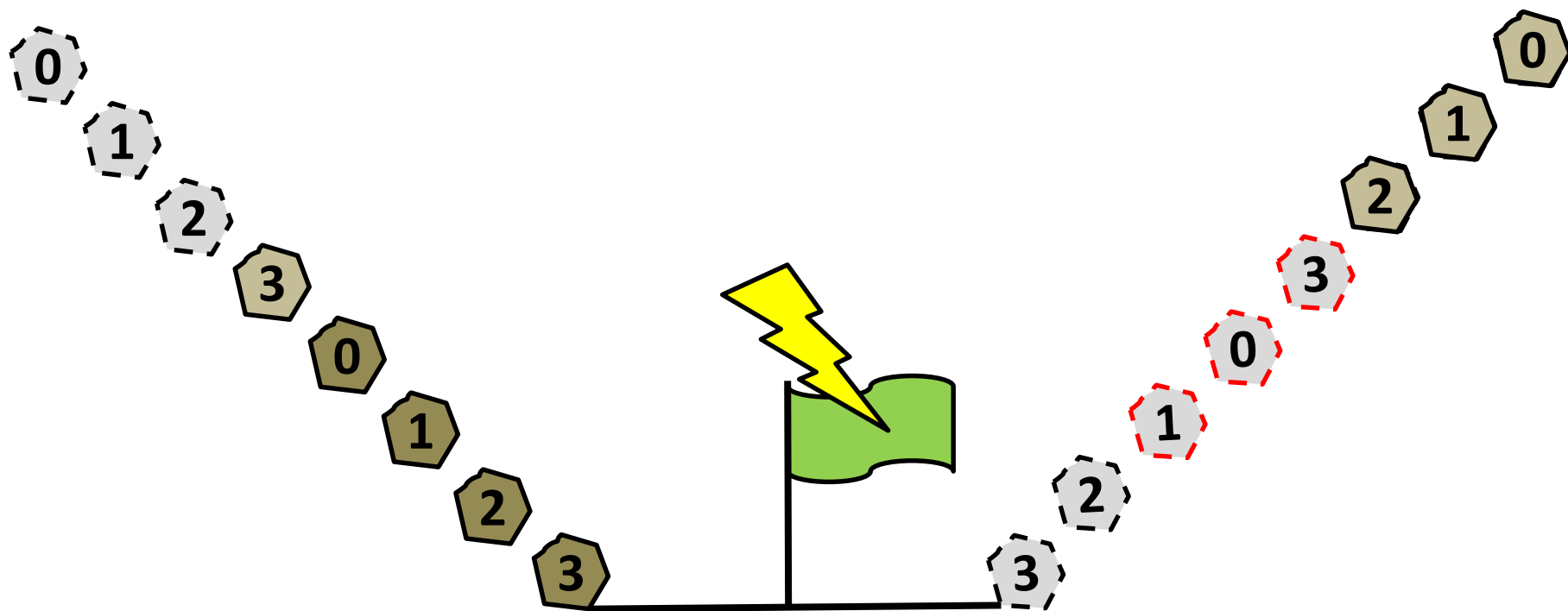
Gyakorló feladat 1. megoldása

- Ha megérkeznének, akkor küldené vissza a nyugtákat.



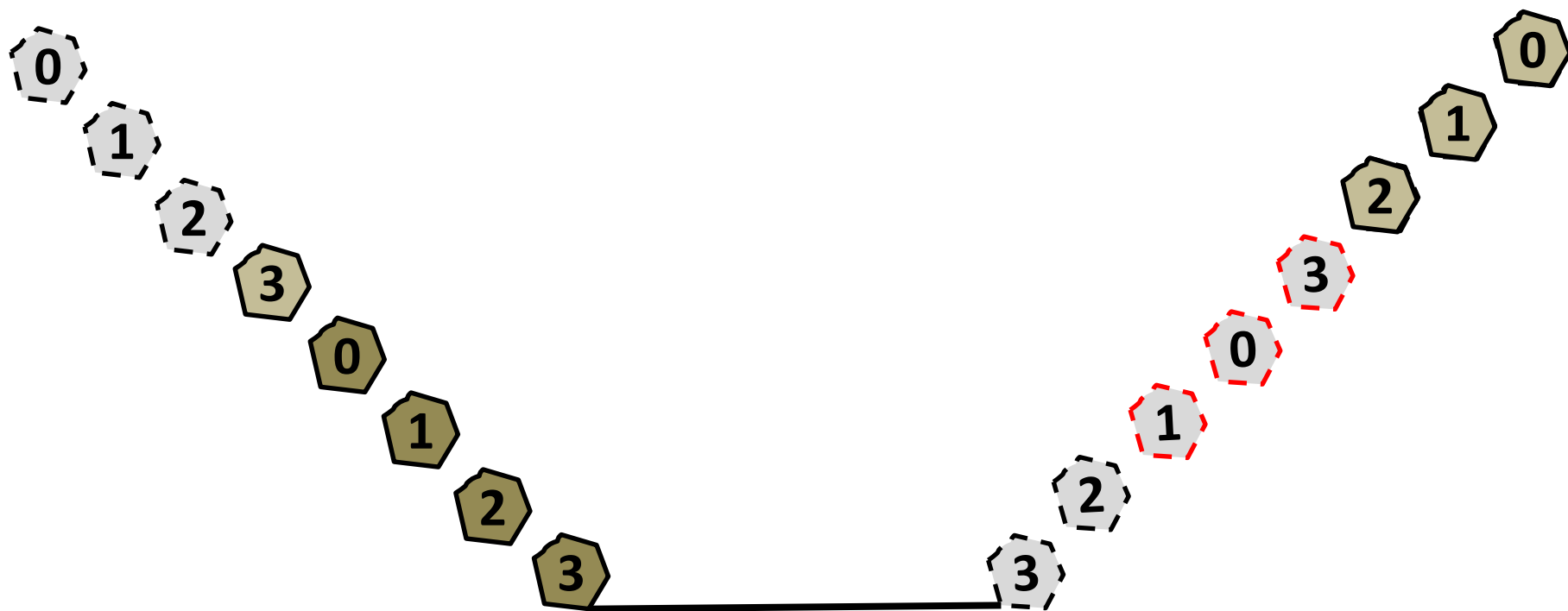
Gyakorló feladat 1. megoldása

- De tegyük fel, hogy azok elvesznének.



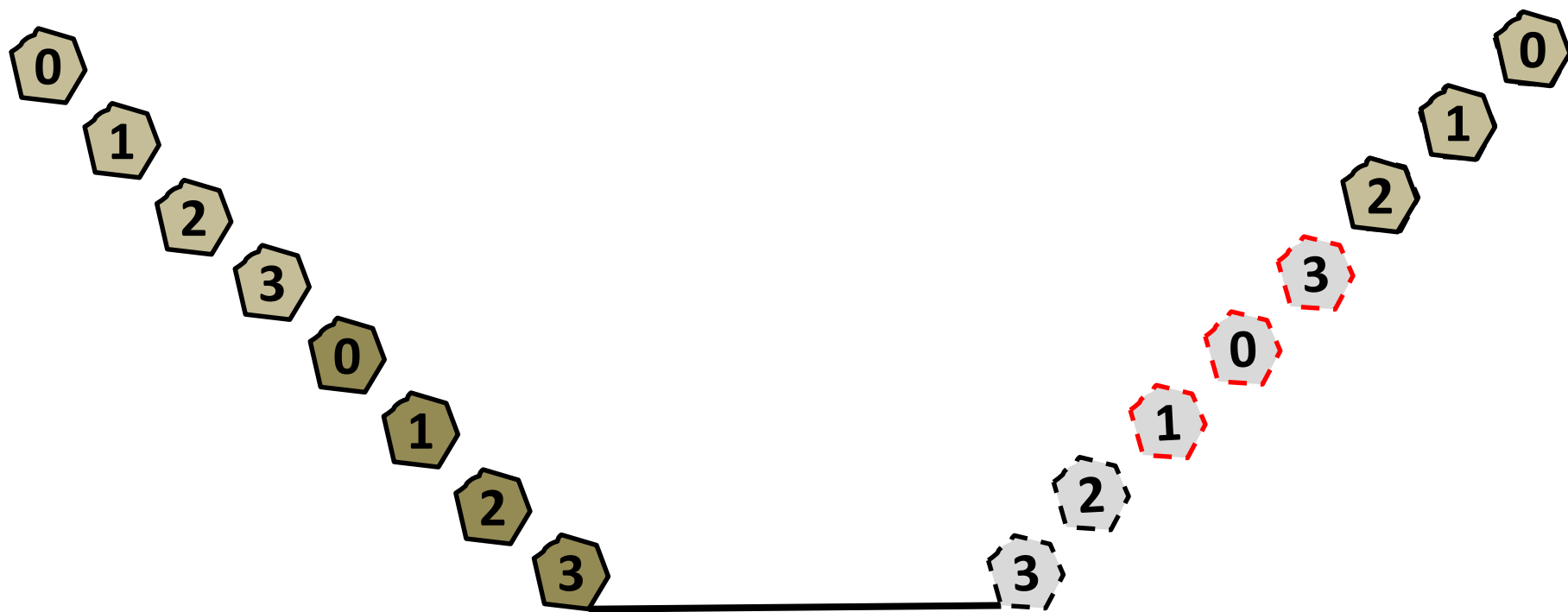
Gyakorló feladat 1. megoldása

- A küldő a timeout után...



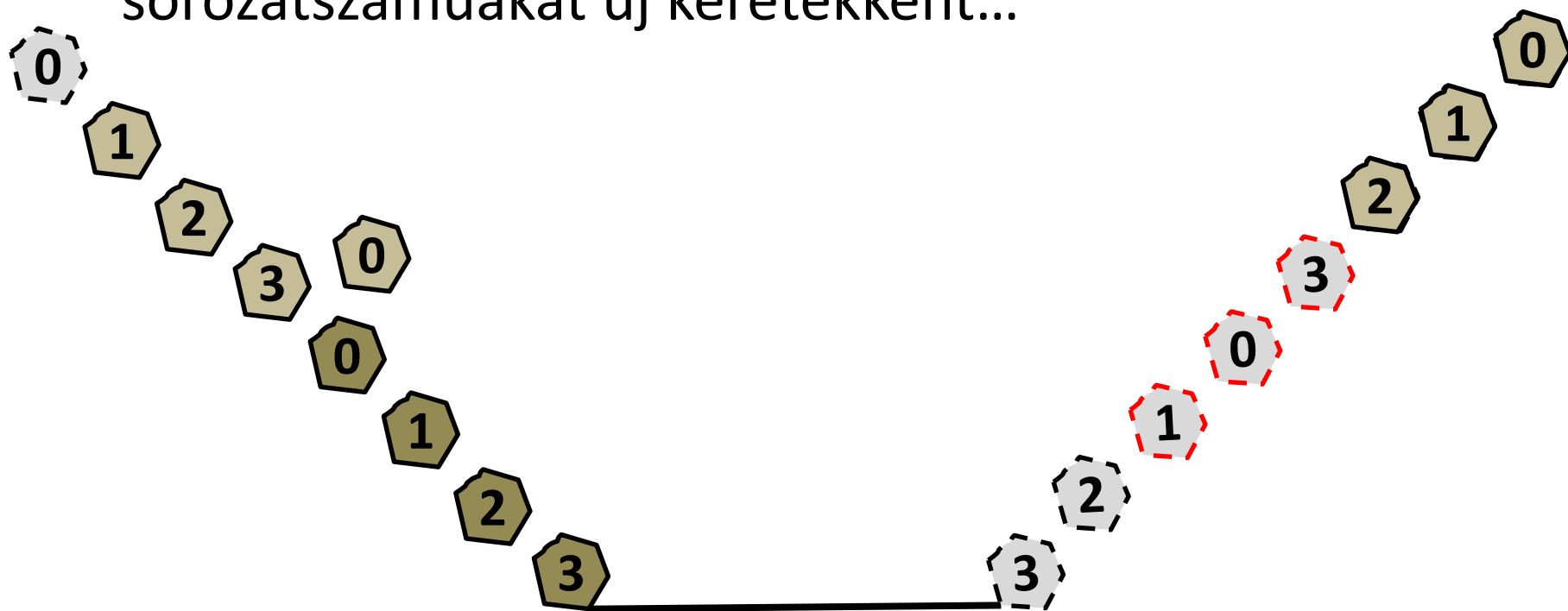
Gyakorló feladat 1. megoldása

- ... újraküldené az első három keretet.



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



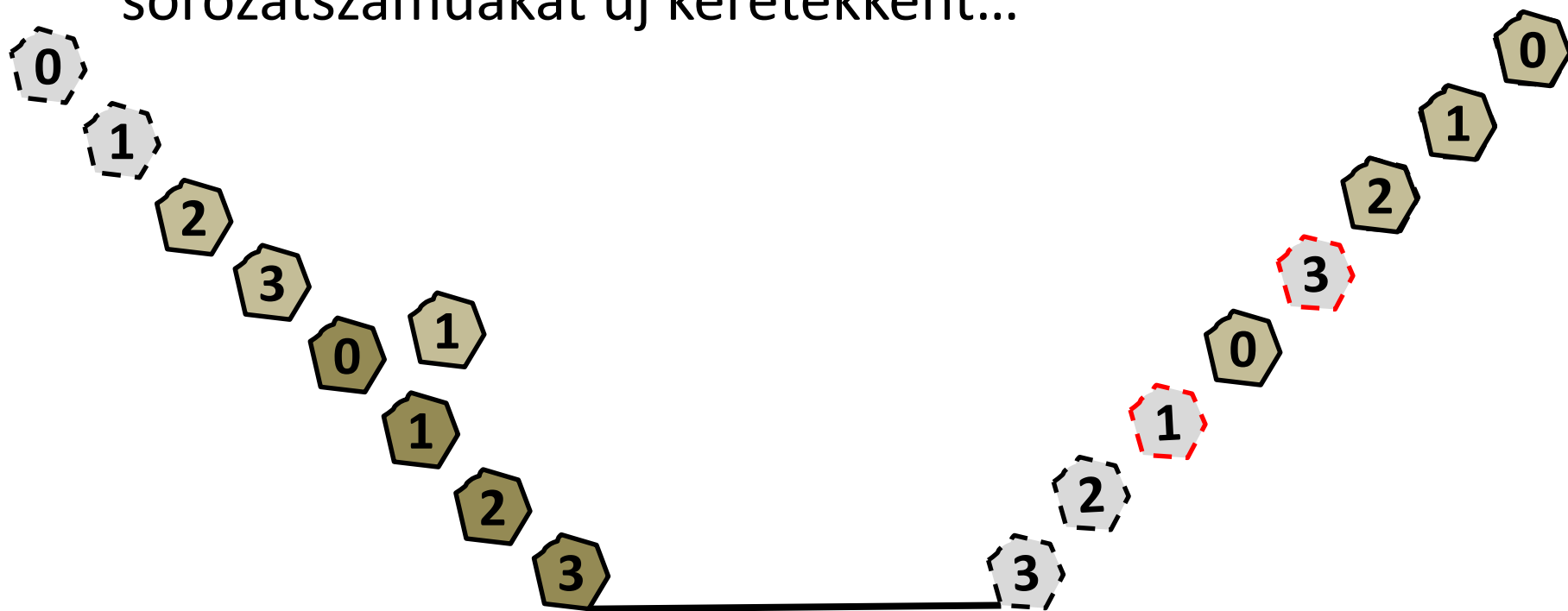
Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



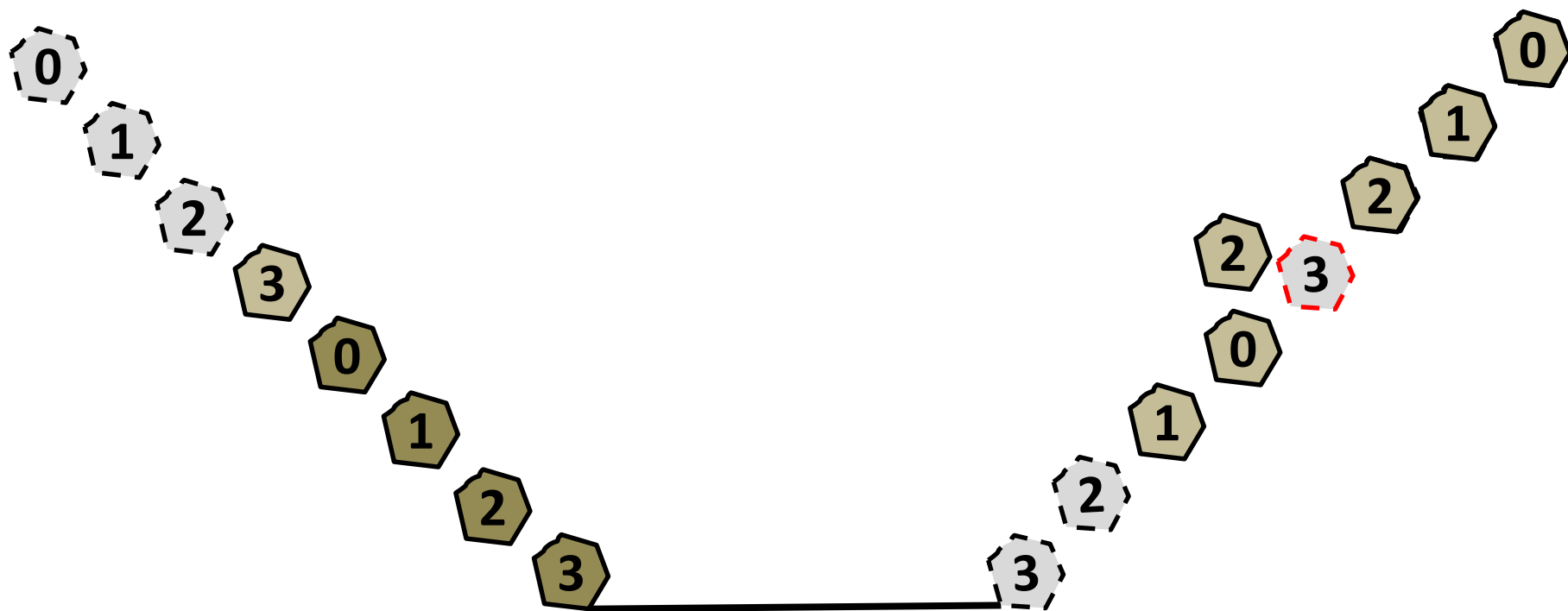
Gyakorló feladat 1. megoldása

- A fogadó feltételezné, hogy ezek már a második sorozatból jönnek, és elfogadná a 0 és 1 sorozatszámúakat új keretekként...



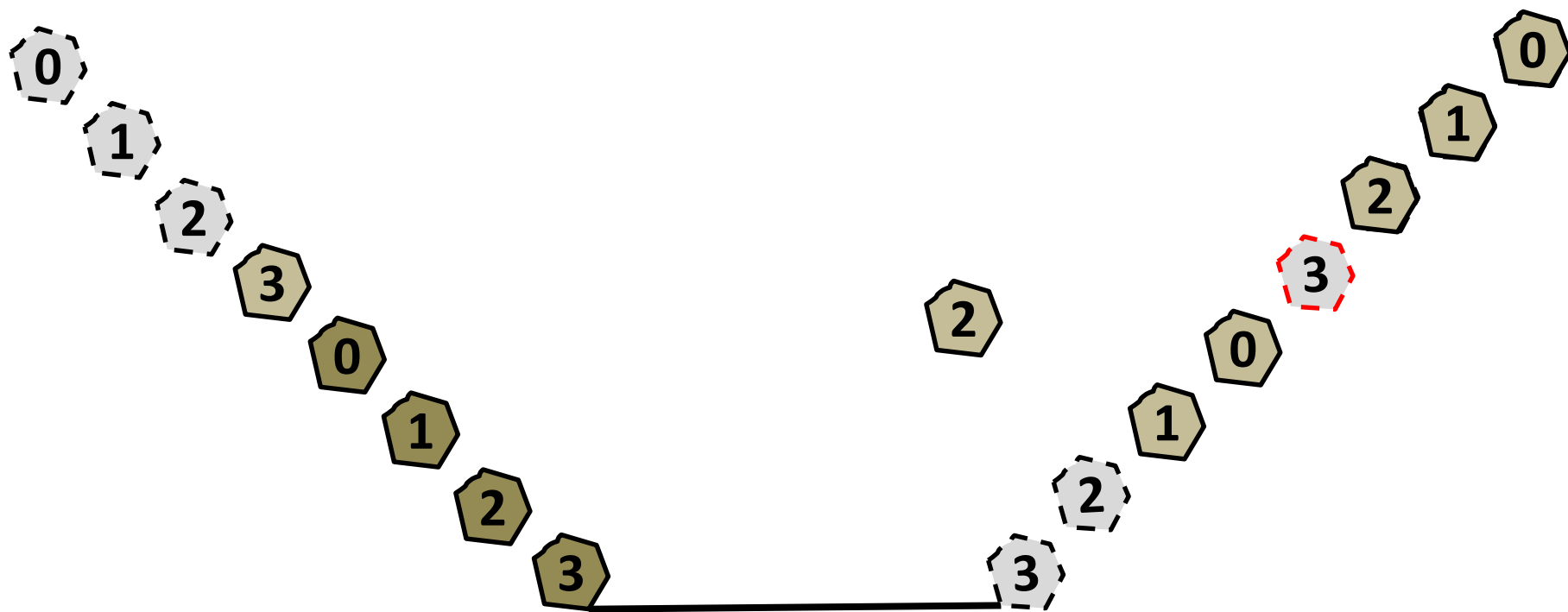
Gyakorló feladat 1. megoldása

- ...de a 2-es sorszámmal küldöttet eldobná, mivel az kívül van a jelenlegi vételi ablakon.



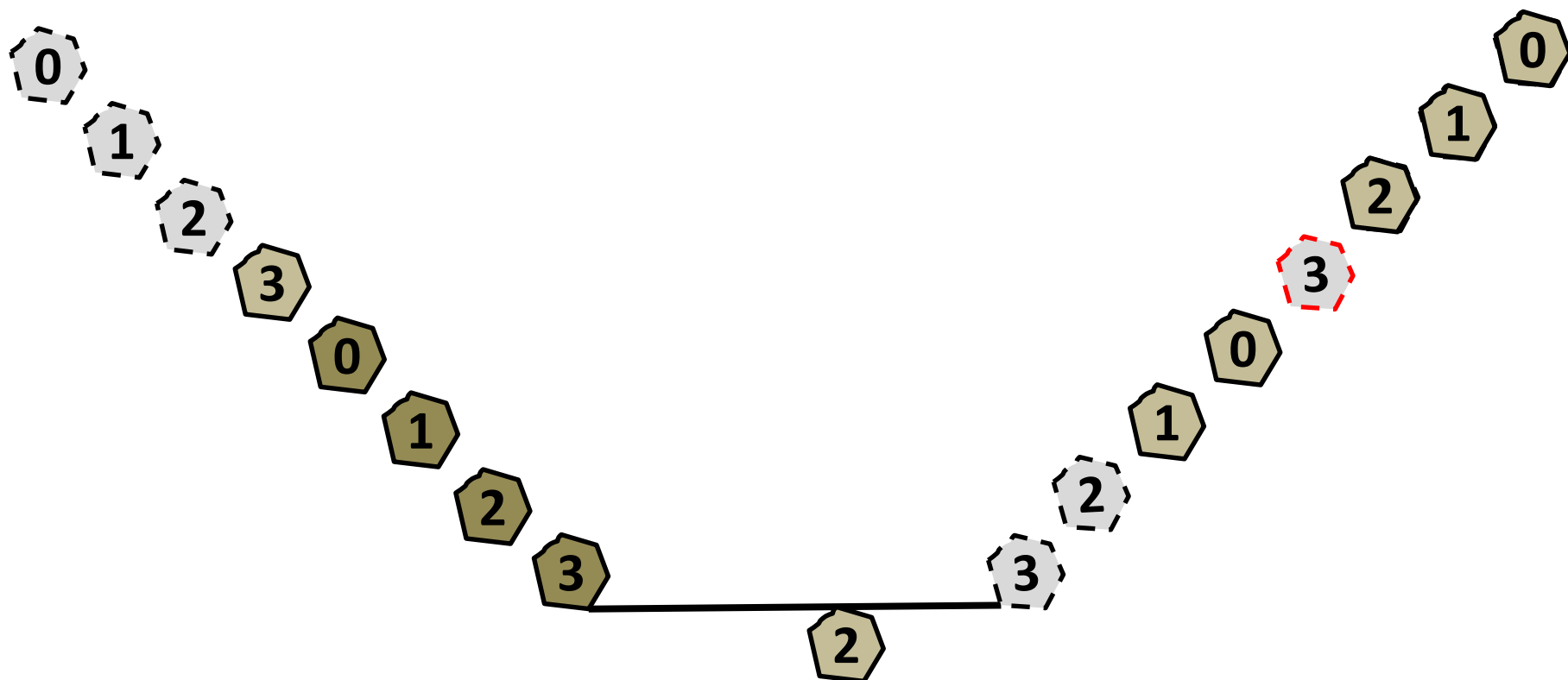
Gyakorló feladat 1. megoldása

- ...de a 2-es sorszámmal küldöttet eldobná, mivel az kívül van a jelenlegi vételi ablakon.

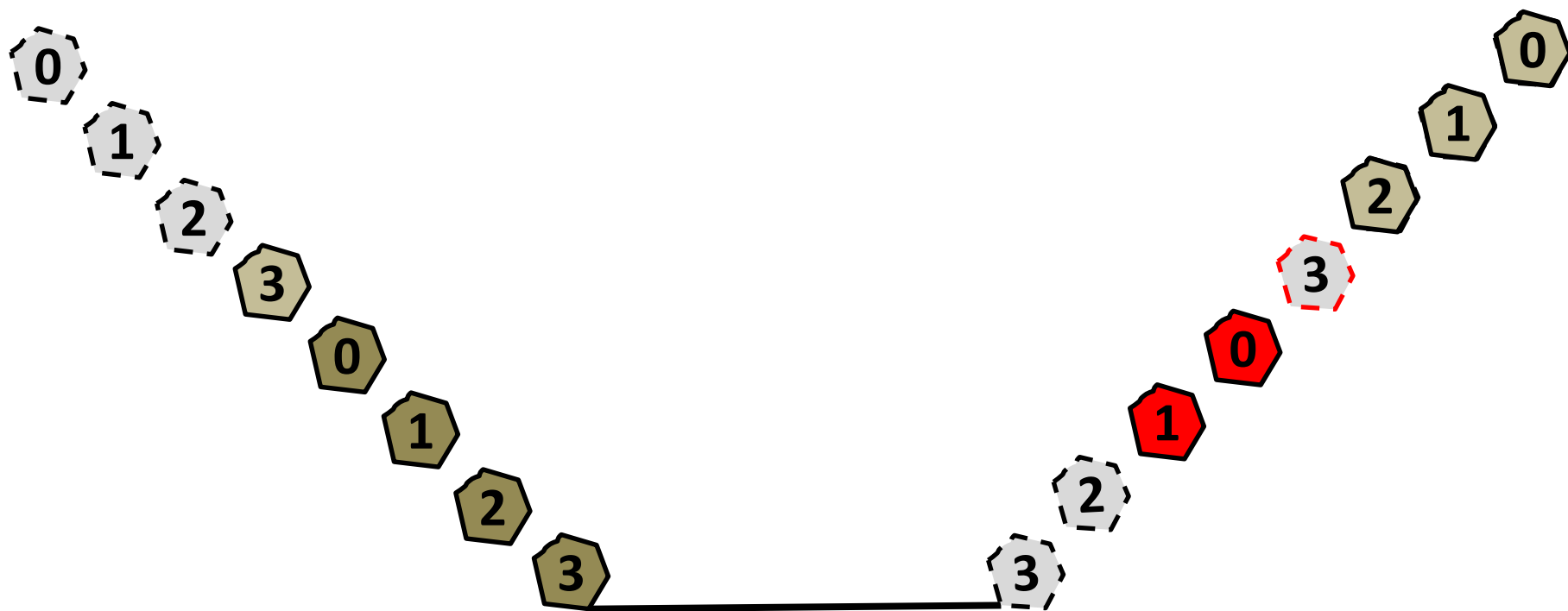


Gyakorló feladat 1. megoldása

- ...de a 2-es sorszámmal küldöttet eldobná, mivel az kívül van a jelenlegi vételi ablakon.



Gyakorló feladat 1. megoldása



MAC alréteg – emlékeztető

- Eddig pont-pont összeköttetés, most adatszóró csatornát használó hálózatok
- A csatorna kiosztás történhet
 - statikus módon (FDM, TDM)
 - N darab felhasználót feltételezünk, a sávszélet N egyenlő méretű sávra osztják, és minden egyes sávhoz hozzárendelnek egy felhasználót.
 - N darab felhasználót feltételezünk, az időegységet N egyenlő méretű időrésre –úgynevezett slot-ra–osztják, és minden egyes réshez hozzárendelnek egy felhasználót.
 - dinamikus módon
 - verseny vagy ütközés alapú protokollok (ALOHA, CSMA, CSMA/CD)
 - verseny-mentes protokollok (bittérkép-alapú protokollok, bináris visszaszámlálás)
 - Korlátozott verseny protokollok (adaptív fa protokollok)

Egy verseny vagy ütközés alapú protokoll: CSMA/CD – emlékeztető

- CSMA/CD (ütközés érzékelő):
 - Minden állomás küldés közben megfigyeli a csatornát,
 - ha ütközést tapasztalna, akkor megszakítja az adást, és véletlen ideig várakozik, majd újra elkezdi leadni a keretét.
- Mikor lehet egy állomás biztos abban, hogy megszerezte magának a csatornát?
 - Az ütközés detektálás minimális ideje az az idő, ami egy jelnek a két legtávolabbi állomás közötti átviteléhez szükséges.

Egy verseny vagy ütközés alapú protokoll: CSMA/CD – emlékeztető

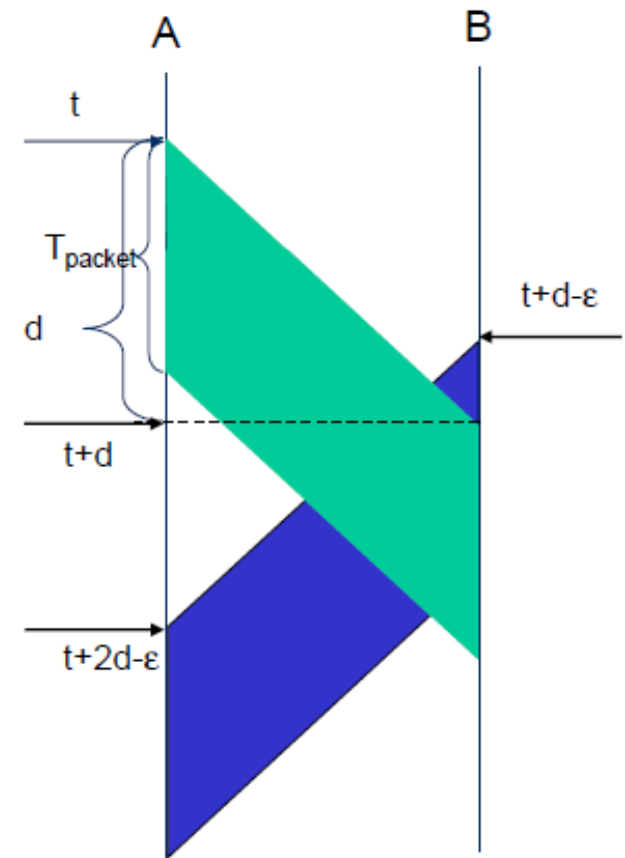
- Ha a két legtávolabbi állomás közötti propagációs késleltetést d_{prop} jelöli, akkor egy keret elküldéséhez legalább $2 \cdot d_{prop}$ idő szükséges.
 - Ahhoz, hogy minden ütközést észleljünk
- A d_{prop} a maximális távolság (l_{max}) és a jel terjedési sebesség (v_{jel}) hányadosa.
- Legyen r a bitráta, és jelölje D_{min} a minimális keretméretet. Ekkor az alábbi összefüggésnek kell fennállnia:

$$D_{min} = r \cdot 2 \cdot d_{prop} = r \cdot 2 \cdot \frac{l_{max}}{v_{jel}}$$

Gyakorló feladat 2.

- Tekintsük egy 1 Gbps CSMA/CD protokoll tervezését maximum 300 méter hosszú rézkábelen való használatra (repeater nincs), melyben az elektromágneses hullámok terjedési sebessége $1.8 \cdot 10^8$ m/s (0,6*fénysebesség). Mekkora a minimális keret méret? Hogyan határozza ezt meg?

- $$D_{min} = r \cdot 2 \cdot d_{prop} = r \cdot 2 \cdot \frac{l_{max}}{v_{jel}}$$

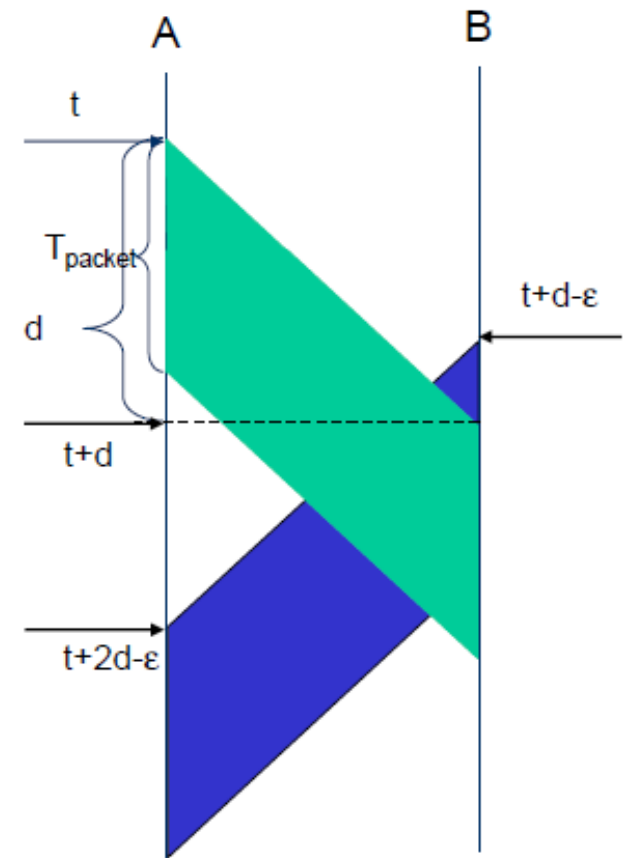


Gyakorló feladat 2. megoldása

- $l_{max} = 300 \text{ m}$, $v_{jel} = 1.8 \cdot 10^8 \text{ m/s}$, $d_{prop} = l_{max}/v_{jel}$
- T_f legyen egy keret elküldéséhez szükséges idő, ekkor egyrészt $T_f = \frac{D_{min}}{r} = \frac{D_{min}}{10^9}$, másrészt $T_f \geq 2 \cdot d_{prop}$:
- $\frac{D_{min}}{10^9} \geq 2 \cdot \frac{l_{max}}{v_{jel}} = 2 \cdot \frac{300}{1.8 \cdot 10^8}$
- $D_{min} \geq \frac{6000}{1.8} = 3333.3333$
- Tehát ahhoz, hogy D_{min} a lehető legkisebb legyen: $D_{min} = 3334$ bitnek kell lenni a keretméretnek.

Gyakorló feladat 3.

- 2 állomás 10 Mbps-es hálózat köt össze közvetlen. tudjuk hogy a minimális keretméret 500 bájt
- - Mekkora lehet a maximális távolság két eszköz között?
- Koax kábelben a sebesség $= 1.8 \cdot 10^8$
- $$D_{min} = r \cdot 2 \cdot d_{prop} = r \cdot 2 \cdot \frac{l_{max}}{v_{jel}}$$



Gyakorló feladat 3. megoldása

- $T_f = \frac{D_{min}}{r} \geq 2 \cdot d_{prop}$
- $500 \text{ bájt} / 10 \text{ Mbps} \geq 2 \cdot d_{prop}$
- $(4 \cdot 10^3 \text{ bit}) / (10^7 \text{ bit/s}) \geq 2 \cdot d_{prop}$
- $2 \cdot 10^{-4} \text{ s} \geq d_{prop} = l_{max} / v_{jel}$
- $2 \cdot 10^{-4} \text{ s} \cdot 1.8 \cdot 10^8 \text{ m/s} \geq l_{max}$
- $36 \text{ km} \geq l_{max}$

Verseny-mentes protokollok – emlékeztető

- Az ütközések hátrányosan hatnak a rendszer teljesítményére
- Az állomások 0-ától N -ig egyértelműen sorszámozva vannak
- Részelt időmodellt feltételezünk

Verseny-mentes protokollok – emlékeztető

- Bináris visszaszámlálás protokoll:
 - Minden állomás azonos hosszú bináris azonosítóval rendelkezik
 - A forgalmazni kívánó állomás elkezd a bináris címet bitenként elküldeni a legnagyobb helyi értékű bittel kezdve
 - Az azonos pozíciójú bitek logikai VAGY kapcsolatba lépnek ütközés esetén
 - Ha az állomás nullát küld, de egyet hall vissza, akkor feladja a küldési szándékát, mert van nála nagyobb azonosítóval rendelkező küldő
 - **Következmény:** a magasabb címmel rendelkező állomásoknak a prioritásuk is magasabb az alacsonyabb című állomásokénál

Verseny-mentes protokollok – emlékeztető

- Mok és Ward módosítása: minden sikeres átvitel után ciklikusan permutáljuk az állomások címét
 - Pl.: ha C, D, A, E, B állomásoknak 4, 3, 2, 1, 0 prioritásai vannak, akkor D sikeres átvitele után D a lista végére fog kerülni,
 - az új prioritás sorrend C, A, E, B, D lesz
 - tehát C továbbra is 4-es prioritású lesz, de a többi megváltozik:
 - D csak akkor fogja tudni használni a csatornát, ha nincs másik állomás, amely akarná

Gyakorló feladat 4.

Szimuláljuk a bináris visszaszámlálás protokollt 8 állomás esetén, ahol az állomás azonosítók rendre a {C, H, D, A, G, B, E, F} halmaz elemei, ez a sorrend a prioritási sorrend is. Ez esetben a virtuális azonosítókat 3 *bit*en ábrázolhatjuk. Tegyük fel, hogy A, C, D és E állomások akarnak egy-egy csomagot átvinni. (Nézze meg mi módosulna a `Mok`- és `Ward`-féle változat esetén.)

Gyakorló feladat 4. megoldása

C – 111, H – 110, D – 101, A – 100, G – 011, B – 010, E – 001, F – 000

idő	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	1	0			1	0	0		1	0	0	adat				
C	1	1	1	adat												
D	1	0			1	0	1	adat								
E	0				0				0				0	0	1	adat

Gyakorló feladat 4. megoldása

(Mok – és Ward)

0. idő:

C – 111, H – 110, D – 101, A – 100, G – 011, B – 010, E – 001, F – 000

4. idő:

H – 111, D – 110, A – 101, G – 100, B – 011, E – 010, F – 001, C – 000

7. kör:

H – 111, A – 110, G – 101, B – 100, E – 011, F – 010, C – 001, D – 000

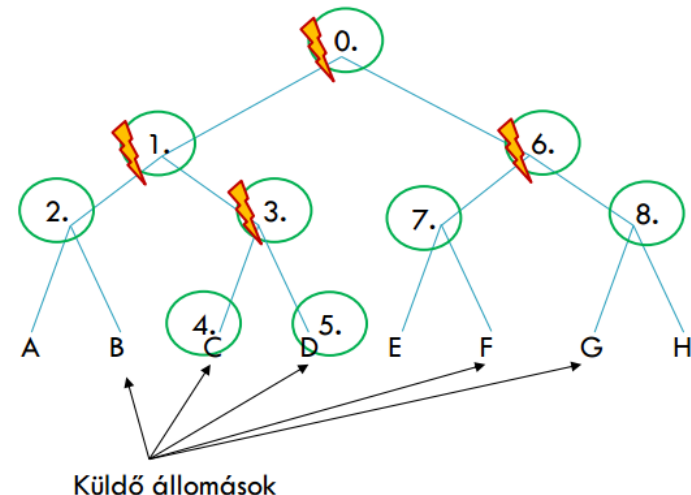
11. kör:

H – 111, G – 110, B – 101, E – 100, F – 011, C – 010, D – 001, A – 000

idő	1	2	3	4	4	5	6	7	8	9	10	11	12	13	14	15	16
A	1	0			1	0			1	1	0	adat					
C	1	1	1	adat													
D	1	0			1	1	0	adat									
E	0				0				0				1	0	0	adat	

Egy korlátozott versenyes protokoll: Adaptív fa bejárás – emlékeztető

- 0-adik időrásben mindenki küldhet
- Ütközés → fa *mélységi bejárása* kezdődik
- Az időrések a fa egyes csomópontjaihoz vannak rendelve
 - Ütközéskor rekurzívan az adott csomópont bal illetve jobb gyerekcsomópontjánál folytatódik a keresés
 - Ha egy időrás kihasználatlan marad, vagy pontosan egy állomás küld, akkor a szóban forgó csomópont keresése befejeződik
 - Például (lásd az ábrán):
 - 0. időrásben: BCDFG
 - 1. időrásben: BCD
 - 2. időrásben: B
 - ...




Egy korlátozott versenyes protokoll: Adaptív fa bejárás – emlékeztető

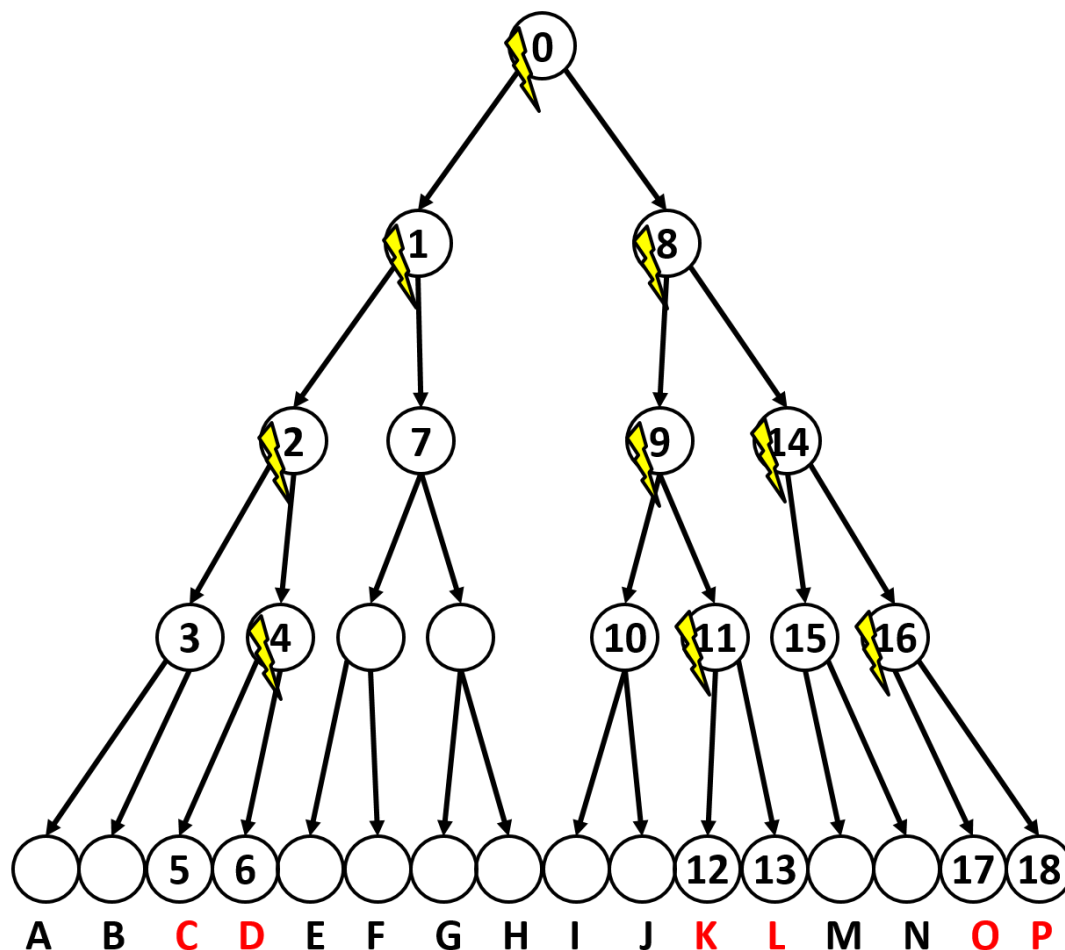
- Adaptív fabejárési protokoll pszeudokódja (<http://classes.uleth.ca/200303/cpsc3780a/chapter4.pdf> alapján)
- Jelölések:
 - $Fa(n)$: az n csomópont alatti részfája, $n.bal$: n csomópont bal oldali gyerekcsomópontja, $n.jobb$: n csomópont jobb oldali gyerekcsomópontja
- **Algoritmus($Fa(n)$):**
 - A $Fa(n)$ -hez tartozó állomások próbálják lefoglalni a csatornát; (Előfordulhat, hogy semelyik (idetartozó) állomás se akar küldeni, ekkor az időrés kihasználatlan lesz.)
 - **if**(nincs ütközés) **return**;
 - $aktuális_időrés++$;
 - **Algoritmus**($Fa(n.bal)$);
 - $aktuális_időrés++$;
 - **Algoritmus**($Fa(n.jobb)$);
- Az algoritmus hívását a gyökértől fogjuk kezdeni:
- $aktuális_időrés:=0$; **Algoritmus**($Fa(root)$);

Gyakorló feladat 5.

- Tekintsünk 16 állomást, melyek adaptív fabejárás protokollal visznek át csomagokat.
- Az állomások azonosítói $\{A, B, C, \dots, O, P\}$.
- Szimulálja a protokoll működését, ha a C, D, K, L, O, P állomások akarnak egy időben csomagot átvinni!
- (Adja meg a verseny időréseket ettől az időpillanattól addig, amíg a protokoll feloldja az ütközést.)

Gyakorló feladat 5. megoldása

- 0. CDKLOP 
- 1. CD 
- 2. CD 
- 3. kihasználatlan
- 4. CD 
- 5. C 
- 6. D 
- 7. kihasználatlan
- 8. KLOP 
- 9. KL 
- 10. kihasználatlan
- 11. KL 
- 12. K 
- 13. L 
- 14. OP 
- 15. kihasználatlan
- 16. OP 
- 17. O 
- 18. P 



Órai / házi feladat

Adaptív fa bejárás

- Az {A, B, C, D, E, F, G, H} **állomások** közül tetszőleges számú versenyeznek a **csatorna** használatért
 - A **csatorna** egy TCP szerver legyen **select** használatával
 - Az **állomások** TCP kliensek
- Ez a gyakorlatban úgy fog kinézni, hogy miután egy **állomás** csatlakozott a **csatornához**, és elküldte az azonosítóját, a *nulladik időrásbe* „kerül”

Órai / házi feladat

Adaptív fa bejárás

- A **csatornán** a *nulladik időrás* „el lesz nyújtva” (pl. 30 másodpercre), hogy a működést szimulálni tudjuk
- Ha a *nulladik időrás*ben többen is küldeni akarnak, tehát ha több **állomással** csatlakoztunk a **csatornához**
→ a csatorna annyit fog visszajelezni a csatlakozott **állomásoknak**, hogy ütközés történt
– (Innentől feltesszük, hogy legalább két **állomás** csatlakozott)
- És kiírja a kimenetre, hogy mely **állomások** akartak volna küldeni ebben az időrásben

Órai / házi feladat

Adaptív fa bejárás

- A **csatorna** ezután a csatlakozott **állomásoknak** elküldi, hogy most következik az *első időérés*, és ez is „el lesz kicsit nyújtva” (pl. 1 másodpercre), ameddig begyűjti azokat állomásokat, akik küldhetnek
- Minden résztvevő **állomás** eldönti az adaptív fa bejárás alapján, hogy küldhet-e az adott időérésben
 - Ha igen → megint elküldi az azonosítóját

Órai / házi feladat

Adaptív fa bejárás

- A **csatorna** jelzi, hogy történt-e ütközés,
- illetve ha nem volt ütközés, és valamelyik **állomás** küldött → visszajelez mindenkinek, hogy sikeres volt az adott **állomás** küldése
- És kiírja a kimenetre, hogy mely **állomások** akartak volna küldeni ebben az időrásben,
- ha nem akart semelyik se küldeni → azt írja ki, hogy „idle”
- Ha egy **állomás**nak sikeres volt a küldése, akkor lezárja a kapcsolatot

Órai / házi feladat

Adaptív fa bejárás

- A **csatorna** ezután a (maradék) csatlakozott **állomásoknak** elküldi, hogy most következik az *második időrés*, és a többi ugyanúgy működik, mint az előbbinél...
- Addig folytatódik, míg az összes résztvevő **állomás** küldése sikeres nem lesz

Órai / házi feladat

Adaptív fa bejárás

- A **csatorna** ezután a (maradék) csatlakozott **állomásoknak** elküldi, hogy most következik az *második időrés*, és a többi ugyanúgy működik, mint az előbbinél...
- Addig folytatódik, míg az összes résztvevő **állomás** küldése sikeres nem lesz
- ***A feladat összetettsége miatt 1 extra pontot lehet szerezni erre a házira, ha minden jól működik***
 - Ha nem tökéletes a program, akkor részpontszámokat továbbra is szerezni lehet

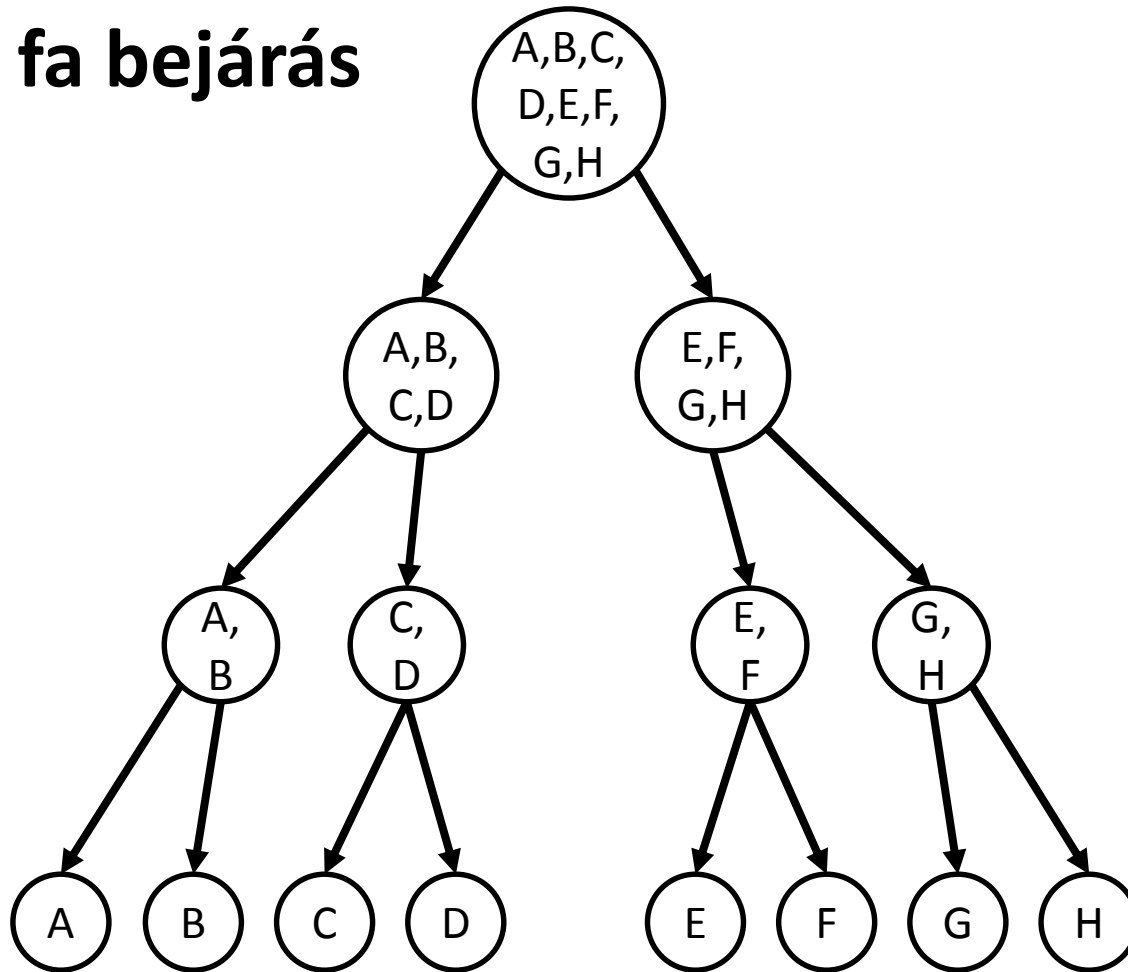
Órai / házi feladat

Adaptív fa bejárás

- Javaslat az adaptív fa elkészítéséhez, bejárásához (de nem kötelező ezt használni):
- Az átláthatóság miatt érdemes lehet előre legyártani a fát úgy, hogy a csomópontjaiban a lehetséges küldő **állomások** szerepeljenek (ld. a következő oldalon az ábrát)
 - (Természetesen csak futás közben fog kiderülni, hogy melyek lesznek ténylegesen ezek közül a küldők)
- A küldő **állomások** a saját adaptív fájukat fogják bejárni a bejövő üzenet alapján
 - Tehát elindulnak a gyökér csomópontból, és ha ütközés történt a *nulladik időrásben* továbbhaladnak a bal részfájuk fele...
 - Ha valamelyik másik **állomás** sikeres volt, vagy kihasználatlan volt egy időrás, akkor felfele kell mozogni majd a fában...

Órai / házi feladat

Adaptív fa bejárás



Órai / házi feladat

Adaptív fa bejárás – 1. futtatás: B, C, D, F, G

```
C:\Windows\System32\cmd.exe
\Gyak8>python haziChannel.py
new connection from ('127.0.0.1', 30025) with stationID G
new connection from ('127.0.0.1', 30026) with stationID B
new connection from ('127.0.0.1', 30027) with stationID D
new connection from ('127.0.0.1', 30028) with stationID C
new connection from ('127.0.0.1', 30029) with stationID F
BCDFG *
BCD *
B OK
CD *
C OK
D OK
FG *
F OK
G OK
```


Órai / házi feladat

Adaptív fa bejárás – 1. futtatás: B, C, D, F, G

```
C:\>\Gyak8>python haziStation.py B
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Close the client
```

```
C:\>\Gyak8>python haziStation.py C
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
collision
Slot 4:
C successful transmission
Close the client
```

```
C:\>\Gyak8>python haziStation.py D
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
collision
Slot 4:
C successful transmission
Slot 5:
D successful transmission
Close the client
```

```
C:\>\Gyak8>python haziStation.py F
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
collision
Slot 4:
C successful transmission
Slot 5:
D successful transmission
Slot 6:
collision
Slot 7:
F successful transmission
Close the client
```

```
C:\>\Gyak8>python haziStation.py G
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
collision
Slot 4:
C successful transmission
Slot 5:
D successful transmission
Slot 6:
collision
Slot 7:
F successful transmission
Slot 8:
G successful transmission
Close the client
```

Órai / házi feladat

Adaptív fa bejárás – 2. futtatás: B, C, G, H

```
C:\Windows\System32\cmd.exe
new connection from ('127.0.0.1', 30030) with stationID G
new connection from ('127.0.0.1', 30031) with stationID B
new connection from ('127.0.0.1', 30032) with stationID C
new connection from ('127.0.0.1', 30033) with stationID H
BCGH *
BC *
B OK
C OK
GH *
idle
GH *
G OK
H OK
```

Órai / házi feladat

Adaptív fa bejárás – 2. futtatás: B, C, G, H

```
C:\>\Gyak8>python haziStation.py G
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
C successful transmission
Slot 4:
collision
Slot 5:
idle
Slot 6:
collision
Slot 7:
G successful transmission

Close the client
```

```
C:\>\Gyak8>python haziStation.py B
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission

Close the client
```

```
C:\>\Gyak8>python haziStation.py C
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
C successful transmission

Close the client
```

```
C:\>\Gyak8>python haziStation.py H
Slot 0:
collision
ABCD
Slot 1:
collision
Slot 2:
B successful transmission
Slot 3:
C successful transmission
Slot 4:
collision
Slot 5:
idle
Slot 6:
collision
Slot 7:
G successful transmission
Slot 8:
H successful transmission

Close the client
```

VÉGE