

Planets Cycles feladatmegoldás

Ez a kód arra szolgál, hogy meghatározza, minden bolygóról indulva hány teleportációra van szükség ahhoz, hogy egy már korábban meglátogatott bolygóra érkezzünk. Minden bolygónak van egy teleportere, amely másik bolygóra (vagy akár önmagára) irányít. Függvénydefiníció:

```
1 def teleportaciok_szamolasa(n, teleporterek):
```

n: a bolygók száma; teleporterek: egy lista, amely megadja, hogy melyik bolygóra mutat az adott bolygó teleportere. Az indexek 0-tól n-ig terjednek, de a teleporterek értékei 1-től n-ig vannak megadva (ezért később korrigáljuk). Eredmény és látogatottság nyilvántartása:

```
2     eredmeny = [0] * n # Az eredményeket tároló lista
3     latogatott = [False] * n # A látogatott bolygókat jelöli
```

eredmeny: egy lista, amelyben minden bolygóhoz eltároljuk a teleportációk számát. Kezdetben minden érték 0; latogatott: egy lista, amelyben minden bolygóról tároljuk, hogy azt már meglátogattuk-e az algoritmus során. Kezdetben minden érték False. Iterálás a bolygók felett:

```
5     for i in range(n):
```

Cél: Végigmenni minden bolygón (i), és elvégezni az algoritmust, ha az aktuális bolygót még nem dolgoztuk fel. Útvonal bejárása:

```
6         if not latogatott[i]:
7             # Új útvonal indítása
8             utvonal = []
9             jelenlegi = i
```

if not latogatott[i]: Csak akkor indítunk bejárást, ha az adott bolygót még nem látogattuk meg; utvonal: Az aktuálisan bejárt bolygók listája. Ez segít a ciklusok és a hozzájuk vezető utak kezelésében; jelenlegi: Az aktuális bolygót jelöli, amelyen éppen állunk. Bolygók bejárása a teleporterek mentén:

```
11         while not latogatott[jelenlegi]:
12             latogatott[jelenlegi] = True
13             utvonal.append(jelenlegi)
14             jelenlegi = teleporterek[jelenlegi] - 1
```

Ciklus feltétele: addig haladunk a teleportereken keresztül, amíg olyan bolygóra érünk, amelyet még nem látogattunk meg. Látogatottság frissítése: az aktuális bolygót megjelöljük látogatottként (latogatott[jelenlegi] = True). Útvonal építése: az aktuális bolygót hozzáadjuk az utvonal listához. Következő bolygó:

az aktuális bolygó teleportere által megadott bolygóra ugunk. A teleporterek 1-indexelt értékét korrigáljuk (-1). Ciklusok kezelése:

```
17         if jelenlegi in utvonal:
18             ciklus_kezdete = utvonal.index(jelenlegi)
19             ciklus_hossza = len(utvonal) - ciklus_kezdete
```

Ciklus felismerése: a jelenlegi bolygó már szerepel az útvonalban, akkor ciklust találtunk. ciklus_kezdete: az utvonal listában az a hely, ahol a ciklus kezdődik; ciklus_hossza: az útvonal hosszából kivonjuk a ciklus kezdőpontjáig tartó út hosszát, így megkapjuk a ciklus hosszát. Eredmények frissítése a ciklusban lévő bolygókra:

```
22         for idx in range(ciklus_kezdete, len(utvonal)):
23             eredmeny[utvonal[idx]] = ciklus_hossza
```

A ciklusban lévő bolygók teleportációs száma megegyezik a ciklus hosszával. Eredmények frissítése a ciklushoz vezető bolygókra:

```
26         for idx in range(ciklus_kezdete):
27             eredmeny[utvonal[idx]] = ciklus_hossza + ciklus_kezdete - idx
```

Azoknál a bolygóknál, amelyek a ciklushoz vezetnek, a teleportációk száma: ciklus_hossza + (ciklus kezdőpontja - aktuális pozíció). Már feldolgozott ciklushoz vezető utak:

```
28     else:
29         # Az eredmény frissítése olyan bolygókra, amelyek egy már kiszámított ciklushoz vezetnek
30         for idx, bolygo in enumerate(utvonal):
31             eredmeny[bolygo] = eredmeny[jelenlegi] + len(utvonal) - idx
```

Ha a jelenlegi bolygó egy már feldolgozott ciklushoz vezet, akkor az útvonal bolygóinak teleportációs számát a cikluson belüli teleportációk számából és a hozzájuk vezető utak hosszából számítjuk ki. Eredmény visszaadása:

```
33     return eredmeny
```

Visszaadjuk az eredmeny listát, amely minden bolygóhoz tartalmazza a teleportációk számát. Főprogram: bemenet olvasása:

```
36     # Bemenet olvasása
37     n = int(input()) # A bolygók száma
38     teleporterek = list(map(int, input().split())) # A teleporterek célpontjai
```

Az első sorban a bolygók száma van megadva (n). A második sorban a teleporterek célpontjai (t_i). Teleportációk számának kiszámítása:

```
40 # Teleportációk számának kiszámítása
41 eredmény = teleportaciok_szamolasa(n, teleporterek)
42
```

Meghívjuk a `teleportaciok_szamolasa` függvényt, hogy kiszámítsa a teleportációk számát. Eredmény kiírása:

```
43 # Eredmény kiírása
44 print(' '.join(map(str, eredmény)))
45
```

Az `eredmeny` listát szóközzel elválasztva kiírjuk a képernyőre. Az algoritmus hatékonyan kezeli a ciklusokat és a hozzájuk vezető utakat.