

CS&ES Problem Set

Planets Cycles

[TASK](#) | [SUBMIT](#) | [RESULTS](#) | [STATISTICS](#) | [TESTS](#) | [QUEUE](#)

Submission details

Task:	Planets Cycles
Sender:	ra3s
Submission time:	2024-12-11 15:54:12 +0200
Language:	Python3 (PyPy3)
Status:	READY
Result:	ACCEPTED

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.04 s	»»
#2	ACCEPTED	0.04 s	»»
#3	ACCEPTED	0.04 s	»»
#4	ACCEPTED	0.04 s	»»
#5	ACCEPTED	0.04 s	»»
#6	ACCEPTED	0.12 s	»»
#7	ACCEPTED	0.10 s	»»
#8	ACCEPTED	0.12 s	»»
#9	ACCEPTED	0.16 s	»»
#10	ACCEPTED	0.15 s	»»
#11	ACCEPTED	0.04 s	»»
#12	ACCEPTED	0.12 s	»»
#13	ACCEPTED	0.04 s	»»
#14	ACCEPTED	0.13 s	»»

Code ▲

```
1 def teleportaciok_szamolasa(n, teleporterek):
2     eredmeny = [0] * n # Az eredményeket tároló lista
3     latogatott = [False] * n # A látogatott bolygókat jelöli
4
5     for i in range(n):
6         if not latogatott[i]:
7             # Új útvonal indítása
8             utvonal = []
9             jelenlegi = i
10
11             while not latogatott[jelenlegi]:
12                 latogatott[jelenlegi] = True
13                 utvonal.append(jelenlegi)
14                 jelenlegi = teleporterek[jelenlegi] - 1
15
16             # Meghatározzuk, hogy találtunk-e ciklust
17             if jelenlegi in utvonal:
18                 ciklus_kezdetek = utvonal.index(jelenlegi)
19                 ciklus_hossza = len(utvonal) - ciklus_kezdetek
20
21                 # Az eredmény frissítése a ciklusban lévő bolygókra
22                 for idx in range(ciklus_kezdetek, len(utvonal)):
23                     eredmeny[utvonal[idx]] = ciklus_hossza
24
25                 # Az eredmény frissítése a ciklushoz vezető bolygókra
26                 for idx in range(ciklus_kezdetek):
27                     eredmeny[utvonal[idx]] = ciklus_hossza + ciklus_kezdetek - idx
28             else:
29                 # Az eredmény frissítése olyan bolygókra, amelyek egy már kiszámolt
30                 # útvonalhoz vezetnek
31                 for idx, bolygo in enumerate(utvonal):
32                     eredmeny[bolygo] = eredmeny[jelenlegi] + len(utvonal) - idx
33
34     return eredmeny
```

Graph Algorithms

...

- Investigation
- Planets Queries I
- Planets Queries II
- Planets Cycles
- Road Reparation
- Road Construction
- Flight Routes Check
- Planets and Kingdoms

...

Your submissions

2024-12-11 15:54:12

2024-12-11 15:52:12

```

33     return erezmeny
34
35
36 # Bemenet olvasása
37 n = int(input()) # A bolygók száma
38 teleporterek = list(map(int, input().split())) # A teleporterek célpontjai
39
40 # Teleportációk számának kiszámítása
41 erezmeny = teleportaciok_szamolasa(n, teleporterek)
42
43 # Erezmeny kiírása
44 print(' '.join(map(str, erezmeny)))

```

[SHARE CODE TO OTHERS](#)

Test details ▲

Test 1

Verdict: **ACCEPTED**

input
10 1 2 3 4 5 6 7 8 9 10

correct output
1 1 1 1 1 1 1 1 1 1

user output
1 1 1 1 1 1 1 1 1 1

Test 2

Verdict: **ACCEPTED**

input
10 1 1 1 1 1 1 1 1 1 1

correct output
1 2 2 2 2 2 2 2 2 2

user output
1 2 2 2 2 2 2 2 2 2

Test 3

Verdict: **ACCEPTED**

input
10 2 3 4 5 6 7 8 9 10 1

correct output
10 10 10 10 10 10 10 10 10 10

user output
10 10 10 10 10 10 10 10 10 10

Test 4

Verdict: **ACCEPTED**

input
10 4 7 8 10 5 7 4 8 3 7

correct output
4 4 2 3 1 4 3 1 3 3

user output
4 4 2 3 1 4 3 1 3 3

Test 5

Verdict: **ACCEPTED**

input
10 6 8 7 2 1 8 6 4 6 4

correct output
5 3 6 3 6 4 5 3 5 4

user output
5 3 6 3 6 4 5 3 5 4

Test 6

Verdict: **ACCEPTED**

input
200000 1 2 3 4 5 6 7 8 9 10 11 12 13 ...

correct output
1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

user output
1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

Test 7

Verdict: **ACCEPTED**

input
200000 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

correct output
1 2 2 2 2 2 2 2 2 2 2 2 2 2 ...

user output
1 2 2 2 2 2 2 2 2 2 2 2 2 2 ...

Test 8

Verdict: **ACCEPTED**

input
200000 2 3 4 5 6 7 8 9 10 11 12 13 14...

correct output
200000 200000 200000 200000 20...

user output
200000 200000 200000 200000 20...

Test 9Verdict: **ACCEPTED**

input
200000 43837 185826 179126 5721 10846...

correct output
803 834 828 804 909 235 104 90...

user output
803 834 828 804 909 235 104 90...

Test 10Verdict: **ACCEPTED**

input
200000 22119 117633 173816 181282 184...

correct output
593 492 506 330 903 365 455 99...

user output
593 492 506 330 903 365 455 99...

Test 11Verdict: **ACCEPTED**

input
3 2 1 1

correct output
2 2 3

user output
2 2 3

Test 12Verdict: **ACCEPTED**

input
200000 2 3 4 5 6 7 8 9 10 11 12 13 14...

correct output
200000 199999 199998 199997 19...

user output
200000 199999 199998 199997 19...

Test 13Verdict: **ACCEPTED**

input
6 2 3 1 5 6 4

correct output
3 3 3 3 3 3

user output
3 3 3 3 3 3

Test 14Verdict: **ACCEPTED**

input
200000 100001 100001 100001 100001 10...

correct output
100001 100001 100001 100001 10...

user output
100001 100001 100001 100001 10...