

Tree Diameter feladatmegoldás

Könyvtárak importálása

```
1 from collections import defaultdict, deque
```

defaultdict: Egy speciális szótár típus, amely alapértelmezett értéket ad vissza, ha egy nem létező kulcshoz férünk hozzá. Ebben az esetben egy listát használunk alapértelmezett értéként, hogy a gráf szomszédsági listáját könnyen építhessük. deque: Egy kétirányú sor, amely hatékonyan támogatja a hozzáadást és eltávolítást mindkét végéről. Ezt használjuk a szélességi keresés (BFS) megvalósításához. Függvény: fa_atmeroje Bemeneti paraméterek:

```
3 def fa_atmeroje(n, elek):
```

n: a csomópontok száma a fában; elek: egy lista, amely az éleket tartalmazza. Minden él két csúcsot ad meg, például (a, b) formában. Szomszédsági lista építése:

```
4     # Szomszédsági lista építése
5     szomszedsag = defaultdict(list)
6     for a, b in elek:
7         szomszedsag[a].append(b)
8         szomszedsag[b].append(a)
```

Szomszédsági lista: Ez a gráf egy reprezentációja, ahol minden csúcsot összekötünk a közvetlen szomszédjaival. BFS függvény:

```
10 def bfs(kezdo pont):
11     tavolsag = [-1] * (n + 1)
12     tavolsag[kezdo pont] = 0
13     q = deque([kezdo pont])
14     legtavolabbi_csucs = kezdo pont
15     while q:
16         jelenlegi = q.popleft()
17         for szomszed in szomszedsag[jelenlegi]:
18             if tavolsag[szomszed] == -1:
19                 tavolsag[szomszed] = tavolsag[jelenlegi] + 1
20                 q.append(szomszed)
21                 if tavolsag[szomszed] > tavolsag[legtavolabbi_csucs]:
22                     legtavolabbi_csucs = szomszed
23     return legtavolabbi_csucs, tavolsag[legtavolabbi_csucs]
```

A BFS (szélességi keresés) végrehajtása egy adott kezdőpontból.

```
10 def bfs(kezdopont):
```

Lépések: távolságok inicializálása:

```
11     tavolsag = [-1] * (n + 1)
12     tavolsag[kezdopont] = 0
```

Minden csomópont távolságát -1-re állítjuk, ami azt jelenti, hogy még nem értük el. A kezdőpont távolságát 0-ra állítjuk. Kezdőpont beállítása a sorba:

```
13     q = deque([kezdopont])
```

A BFS sor (q) tartalmazza a kezdő csúcsot. Legtávolabbi csúcs inicializálása:

```
14     legtavolabbi_csucs = kezdopont
```

Kezdetben a kezdőpont a legtávolabbi csúcs. BFS ciklus:

```
15     while q:
16         jelenlegi = q.popleft()
```

A ciklus mindaddig folytatódik, amíg a sor nem üres. Az aktuális csúcsot kivesszük a sor elejéről. Szomszédok bejárása:

```
17         for szomszed in szomszedsag[jelenlegi]:
18             if tavolsag[szomszed] == -1:
19                 tavolsag[szomszed] = tavolsag[jelenlegi] + 1
20                 q.append(szomszed)
```

Minden szomszédot ellenőrzünk, amelyet még nem látogattunk meg (távolsága -1). A távolságát beállítjuk az aktuális csúcs távolsága +1-re. A szomszédot hozzáadjuk a sorhoz. Legtávolabbi csúcs frissítése:

```
21             if tavolsag[szomszed] > tavolsag[legtavolabbi_csucs]:
22                 legtavolabbi_csucs = szomszed
```

Ha egy szomszéd távolsága nagyobb, mint az eddig ismert legtávolabbi csúcs távolsága, akkor frissítjük a legtávolabbi csúcsot. Visszatérési érték:

```
23     return legtavolabbi_csucs, tavolsag[legtavolabbi_csucs]
```

Visszaadjuk a legtávolabbi csúcsot és a távolságát. Kétszeri BFS a fa átmérőjének meghatározásához:

```
26     csucs1, _ = bfs(1)
27
28     # Második BFS a fa átmérőjének meghatározásához
29     _, atmero = bfs(csucs1)
```

Első BFS: az első BFS egy véletlenszerű csúcsból indul (például 1-ből), hogy megtaláljuk a fa egyik végpontját. Ez a végpont a fa egyik legszélső csúcsa lesz, amit nevezzünk csucs1-nek. Második BFS: a második BFS a csucs1 csúcsból indul. Ez határozza meg a fa átmérőjét, amely a csucs1-től legtávolabbi csúcs távolsága. Átmérő visszaadása:

```
31     return atmero
```

A függvény visszaadja a fa átmérőjét. Bemenet beolvasása:

```
33     # Bemenet beolvasása
34     n = int(input())
35     elek = [tuple(map(int, input().split())) for _ in range(n - 1)]
36
```

Az első sorban beolvassuk a csomópontok számát (n). A következő n-1 sorban beolvassuk a fa éleit. Fa átmérőjének meghatározása:

```
37     # Fa átmérőjének meghatározása
38     print(fa_atmeroje(n, elek))
```

Meghívjuk a fa_atmeroje függvényt az élek és a csomópontok számának megadásával, majd kiírjuk az átmérőt.