

Podan je slovar postaje, katerega ključi so imena postaj Biciklja, vrednosti pa slovarji z zemljepisnimi koordinatami postajališč, kapaciteto (število stojal) in trenutnim številom koles na postaji. (Glej slovar v testih.)

Podana je tudi funkcija razdalje(postaja1, postaja2), ki prejme imeni postaj in vrne razdaljo med njima.

1. Branje podatkov

Napišite funkcijo prenesi(), ki prebere podatke z naslova

<https://api.jcdecaux.com/vls/v3/stations?apiKey=frifk0jbxfeqgnigez09tw4jvk37wyf823b5j1i&contract=ljubljana> v času izpita pa z <https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=59478> (naslov je shranjen v spremenljivki URL v testih), jih predela v slovar v enaki obliki kot podani slovar postaje, in ga vrne.

2. Najbližje 3

Avtor izpita ni uporabnik Biciklja (enkrat je bilo dovolj), ve pa, da na določeni postaji včasih ne moreš dobiti ali oddati kolesa, zato se je potrebno ozreti po bližnjih postajah. Napišite funkcijo najblizje3(postaja), ki za podano postajo poišče najbližje tri postaje in vrne seznam (treh) parov (razdalja, ime). Seznam naj bo urejen po razdaljah.

Klic najblizje3("ŽIVALSKI VRT") vrne [(0.73687236710574, 'VIŠKO POLJE'), (0.9639016462683163, 'TEHNOLOŠKI PARK'), (1.0760703913614853, 'SOSESKA NOVO BRDO')].

3. Tabela postaj

Napišite funkcijo zapisi(ime_datoteke), ki v podano datoteko zapiše imena postaj (na 40 mest), ki ji sledi število razpoložljivih koles in število mest. Zapis mora biti urejen po abecedi imen postaj (šumniki pa bodo na koncu; uporabite običajno urejanje v Pythonu) in v obliki, ki jo kaže okvirček.

KONGRESNI TRG-ŠUBIČEVA ULICA	9/20
KOPALIŠČE ILIRIJA	0/20
KOPALIŠČE KOLEZIJA	19/20
KOPRSKA ULICA	3/8
KOSEŠKI BAJER	14/20

4. Prevozi

Imamo matriko v numpyju, ki predstavlja podatke o prehodih koles med postajami v nekem časovnem obdobju: element z indeksoma [i][j] pove, koliko koles je bilo prevzetih na i-ti postaji in vrnjenih na j-to. Matrika na sliki nam pove, da so na postaji 1 prevzeli 3 kolesa, ki so jih vrnili na postajo 1; 7 koles, ki so jih vrnili na postajo 2; in 5 koles, ki so jih vrnili na 4.

```
[[1, 2, 1, 0, 3, 0],
 [0, 3, 7, 0, 5, 0],
 [0, 1, 0, 2, 3, 0],
 [0, 3, 0, 0, 4, 2],
 [0, 1, 5, 0, 0, 3],
 [0, 0, 5, 1, 3, 0]]
```

- Skupno so s postaje 1 odpeljali 15 koles, pripeljali pa so jih (kje to vidimo?) 10. Na postaji 1 je pet koles manj, kot jih je bilo prej. Napišite funkcijo spremembe(prevozi), ki prejme takšno matriko in vrne matriko s toliko elementi, kolikor je postaj. Vsak element pove, koliko koles več (pozitivna števila) oz. manj (negativna) kot prej je na postaji. Za matriko s slike vrne [-6, -5, 12, -6, 9, -4].
- S postaje 1 na postajo 4 so pripeljali 5 koles, s postaje 5 na 1 pa samo 1 kolo. Razlika je torej 4. S postaje 4 na 5 so jih pripeljali 3, obratno prav tako 3. Razlika je 0. Največja razlika je med postajama 1 in 2: v eno smer so jih pripeljali 1, v drugo 7, razlika je 6. Napišite funkcijo asimetrija(prevozi), ki vrne par postaj z največjo razliko. V gornjem primeru mora vrniti [1, 2] (ali [2, 1]). Če je parov z največjo razliko več, smete vrniti poljubnega.

Zaželeno je, da nalogo rešite s spretno uporabo numpyja, brez kakšnih zank v Pythonu.

5. Spusti

Po nekem strašnem naključju predstavlja matrika na gornji sliki tudi nadmorske višine nekih točk na neki mreži. Kolesar vozi le navzdol: če se znajde na polju [1][1] (trojka) lahko gre gor (na 2), levo (na 0) ali dol (na 1), ne pa desno, ker je tam 7. Po diagonalah ne vozi.

Napišite funkcijo spust_na_0(matrika, x, y), ki za podani x (stolpec) in y (vrstica) pove, koliko ničel je dosegljivih s tega polja. Če je možno do neke ničle priti na več načinov, jo štejte večkrat, ker je kolesar neumen in ne opazi, kadar pride do istega cilja samo po drugi poti.

```
[[1, 2, 1, 0, 3, 0],
 [0, 3, 7, 0, 5, 0],
 [0, 1, 0, 2, 3, 0],
 [0, 3, 0, 0, 4, 2],
 [0, 1, 5, 0, 0, 3],
 [0, 0, 5, 1, 3, 0]]
```

Z omenjene trojke lahko pride na petih ničele: na **eno** neposredno, do **dveh** prek spodnje enke, če gre gor na dvojko, pa lahko gre od ondod levo ali desno na enki v prvi vrstici in potem na ničli zraven njiju. Ničlo v prvi vrstici štejemo dvakrat, ker jo dosežemo na dva načina.

Nasvet: ne vznemirjajte se zaradi "več načinov": naloga je takšna zato, da je lažja, ne težja. Če se boste "normalno" lotili naloge, boste večkratne poti že "ponesreči" šteli večkrat.