

V primeru, da marsovske invazije ne bomo ustavili v zraku, se bo treba pripraviti na kopenske akcije.

Večina funkcij, ki jih boste morali napisati, prejema dva argumenta:

- seznam povezav med kraji, na primer [("Piran", "Koper"), ("Koper", "Postojna"), ("Postojna", "Logatec")]. Vrstni red elementov ni pomemben; je možno iti iz Pirana v Koper in iz Kopra v Piran;
- slovar višin, katerega ključi so imena krajev, pripadajoče vrednosti pa njihove nadmorske višine.

Večina testov uporablja povezave, narisane na zemljevidu **na drugi strani pole**.

Nasvet: napiši si pomožno funkcijo, ki prejme seznam povezav in vrne slovar, katerega imena so kraji, vrednosti pa vsi kraji, povezani s tem krajem, npr. {"Piran": {"Koper"}, "Koper": {"Piran", "Postojna"}, "Postojna": {"Koper", "Logatec"}, "Logatec": {"Postojna"}}. Prav ti bo prišla pri dveh ali treh funkcijah.

1. Ogrožena mesta

Napiši funkcijo `ogrozena(povezave)`, ki vrne množico ogroženih mest, ki so povezana z enim samim mestom. Za primer z zemljevida vrne {"Piran", "Ilirska Bistrica", "Gornja Radgona"}. Ker bomo v zemljevid morda vključili vsa slovenska naselja, mora funkcija delovati tudi, če krajev zelo veliko.

2. Varna mesta

Napiši funkcijo `varna(povezave)`, ki vrne seznam varnih mest. Mesto je varno, če nastopa v trikotniku med sabo povezanih mest. Primera takih trikotnikov sta Logatec, Idrija in Žiri ter Ljubljana, Kamnik in Kranj.

3. Možna pot

Ker je gravitacija na Marsu nižja, ugibamo, da marsovska pehota ne bo mogla iz kraja z nižjo v kraj z višjo nadmorsko višino. Iz Cerknice lahko gredo na morje (Cerknica – Postojna – Koper), ne pa na Ptuj, ker ne obstaja pot iz Cerknice do Ptuja, na kateri se ne bi nekje povzpeli. (Vmesne klance zanemarimo; med Postojno in Kopro se tudi dvignemo, a na zemljevidu tega ni.) Napiši funkcijo `mozna_pot(odkod, kam, povezave, visine)`, ki vrne `True`, če obstaja nenaraščajoča pot odkod kam in `False`, če je ni.

4. Zasede

Takoj, ko marsovska enota prehodi neko povezavo, postavimo nanjo zasedo, ki bo ustavila vse naslednje enote, ki bi hotele uporabiti isto pot. Zato lahko Marsovci vsako pot uporabijo le enkrat. Recimo, da pošljejo tri enote z načrti:

```
poti = [{"Postojna", "Logatec", "Idrija", "Bled", "Kranj"}
        ["Jesenice", "Bled", "Kranj", "Kamnik"]
        ["Celje", "Kamnik", "Kranj", "Ljubljana"]]
```

Enote gredo na pot ena za drugo, ne hkrati. Na cilj bo prišla le druga. Prva se ustavila v Idriji, ker ni povezave med Idrijo in Bledom, tretja pa v Kamniku, ker je povezavo Kamnik – Kranj (sicer v nasprotni smeri) uporabila že druga enota. Druga pride na cilj, ker, ne spreglej, sme iti po poti Bled – Kranj, ker prva ni šla po njej, saj se je ustavila v Idriji.

Predpostavka, da marsovci ne morejo hoditi navkreber, je bila optimistična, zato bomo v tej nalogi višine zanemarili.

Napiši funkcijo `zasede(poti, povezave)`, ki dobi seznam poti v takšni obliki in vrne število enot, ki bodo prišle na cilj.

5. Branje podatkov

Zemljevid je shranjen v datoteki. Ta vsebuje vrstice z imenom kraja, dvopičjem in nadmorsko višino. Sledi prazna vrstica, nato povezave: ime kraja, dvopičje in kraji, povezani z njim. Kraji so ločeni z vejicami.

Napiši funkcijo `zemljevid(ime_datoteke)`, ki vrne slovar višin in seznam povezav v oblikah, ki smo ju vajeni.

```
Ilirska Bistrica: 440
Postojna: 555
Nova Gorica: 93
Ribnica: 492
Logatec: 481
Cerknica: 559
```

```
Logatec: Cerknica
Cerknica: Ribnica, Postojna
Postojna: Nova Gorica, Logatec, Ilirska Bistrica
```

