Izpit iz predmeta Programiranje 1, 18. januar 2023, 12.00 **Čas pisanja: ~~60 minut~~ → 90 minut**
Dovoljena je uporaba literature (Učilnica in vse, kar prinesete s seboj), prepovedana pa je vsakršna komunikacija.

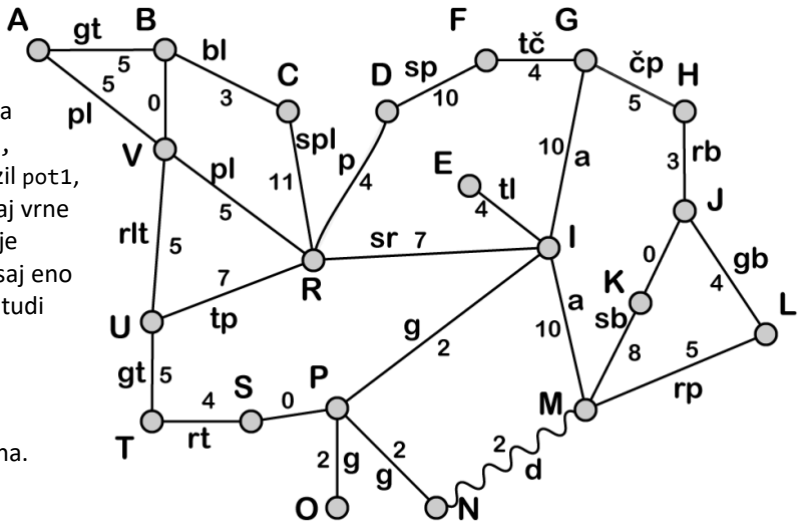
1. Skupni odsek

Dva kolesarja sta šla istočasno na pot. Za vsako povezavo sta potrebovala enako časa. Napiši funkcijo `skupni_odsek(pot1, pot2)`, ki vrne dolžino najdaljšega zaporedja, na katerih sta bila skupaj. Poti sta podani z nizom zaporednih črk, ki označujejo križišča. Za primer v okvirčku vrne 5, saj je najdaljše skupno zaporedje TUMVI.

```
ASAIMWGVIEMHEUTUMVIVHIV
OIMIMAWAREMMPBTUMVIBTIOWE
```

2. Večji frajer

Kolesarja sta na svoji poteh po Ljubljani razkazovala veščine. Napiši funkcijo `vecji_frajer(pot1, pot2, zemljevid)`, ki vrne `True`, če je kolesar, ki je prevozil `pot1`, večji frajer od kolesarja, ki je prevozil `pot2`. Sicer naj vrne `False`. Prvi je večji frajer, če je vsako veščino, ki jo je uporabil drugi, uporabil vsaj tolikokrat kot drugi, vsaj eno veščino pa večkrat kot drugi (pri čemer je to lahko tudi neka veščina, ki je drugi sploh ni pokazal).



Ključi slovarja `zemljevid` so pari (`tuple`) križišč, pripadajoče vrednosti pa množica veščin, ki jih je potrebno uporabiti na povezavi. Pot je vedno možna.

3. Branje zemljevida

Zemljevid je shranjen v datoteki v takšni obliki, kot jo kaže slika na desni. Napiši funkcijo **`preberi_zemljevid_vescin(ime_datoteke)`**, ki prejme ime datoteke z zemljevidom in vrne slovar, katerega ključi so povezave (par križišč), pripadajoče vrednosti pa veščine, zahtevane za to povezavo. Za primer na sliki mora vrniti `{("BF", "FRI"): {"trava", "gravel", "pesek"}, {("BF", "FDV"): {"pesek"}}, {("FRI", "EF"): {"trava"}}`.

```
trava: BF-FRI, FRI-EF
gravel: BF-FRI
pešci:
pesek: BF-FRI, BF-FDV
```

4. Izbire

Ker je prva prioriteta Mestne občine Ljubljana (MOL) varnost kolesarjev, so sprejeli predpis, po katerem smemo iz vsakega križišča voziti le v križišča, katerih ime je po abecedi kasnejše od trenutnega: iz D smemo v R, obratno pa ne.

Kolesarji se, kot vedno, pritožujejo, zato bi MOL rad pokazal, da bodo kljub tej omejitvi še vedno lahko pokazali vse, kar znajo. Sestavi funkcijo **`koristne_vescine(tocka, zemljevid)`**, ki vrne množico vseh veščin, ki lahko pridejo prav nekomu, ki začne v podani točki. Klic `koristne_vescine("P", zemljevid)` vrne `{'gravel', 'trava', 'robnik', 'lonci'}`, saj so to vse veščine, ki se pojavijo, če začnemo v P in gremo po poljubni poti, ki pa nikoli ne gre v točko, ki je po abecedi pred trenutno.

5. Kolesar - frajer

Vaši kolegi so ob devetih napisali razred `Kolesar` z naslednjimi metodami. **Ta razred je že napisan in priložen testom.**

- Konstruktor prejme ime začetne točke, zemljevid in množico veščin, ki jih kolesar obvlada.
- Metoda `premik(kam)` ga premakne s trenutne točke na podano, vendar le, če obstaja povezava s trenutne točke do podane in kolesar obvlada potrebne veščine. Sicer pa zleti s ceste in se ne premakne nikoli več. (Sam si je kriv: pa naj se nauči skakati čez robnike, če hoče voziti po ljubljanskih kolesarskih stezah!
- Metoda `lokacija()` vrne trenutno lokacijo ali `None`, če kolesar leži pod cesto.

Naloga je, da iz njega izpeljete razred Frajer.

- `Frajer` ima tri življenja. Če poskusi premik, ki ni možen, se ne premakne, vendar se lahko ob naslednjem klicu metode premik vseeno premika naprej. Ampak samo dvakrat. Ko izgubi tretje življenje, dokončno obleži.
- Metoda zivljenja() vrne število preostalih življenje.
- Metoda `uporabljene()`, ki vrne množico veščin, ki jih je kolesar pokazal na svoji poti.

Razred izpelji tako, da bo imel svojo metodo `premik`, ki pa bo za samo premikanje poklicala podedovano metodo premik. Prav tako naj morebitni novi konstruktor pokliče podedovani konstruktor.