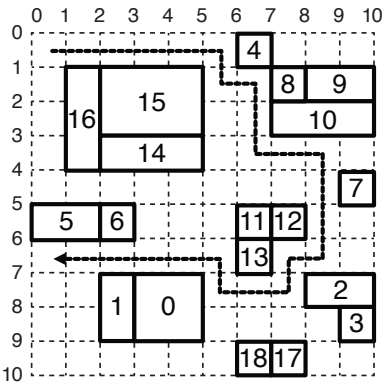
Ovire so predstavljene s četverko `(x0, y0, x1, y1)`, ki predstavlja koordinato gornjega levega in spodnjega desnega oglišča. Ovira 15, recimo, je predstavljena z (2, 1, 5, 3). Ovire nikoli ne sovpadajo. Kadar so ovire podane s seznamom so v večini testov oštevilčene, kot kaže slika.



Kvadratki so označeni tako, da ima levi gornji kvadratek koordinate (0, 0), tisti desno od njega je (1, 0) in tako naprej.

V datoteki s testi vam je podarjena funkcija `se_dotikata(ovira1, ovira2)`, ki vrne `True`, če se podani oviri dotikata – četudi samo po diagonali, kot 12 in 13.

Ni prepovedano, da si napišete funkcijo `poisci_oviro(x, y, ovire)`, ki vrne oviro, ki vsebuje točko (x, y), oziroma `None`, če takšne ovire ni.

1. Pot

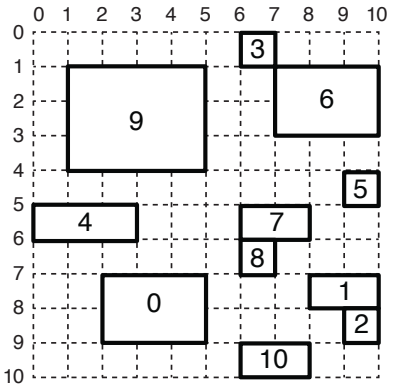
Napiši funkcijo `mozna_pot(x, y, pot, ovire)`, ki prejme začetni koordinati kolesarja, pot v obliki niza (glej spodnji primer) in množico četverk, ki predstavljajo ovire. Funkcija vrne `True`, če lahko kolesar prevozi podano pot, na da bi se zaletel v oviro. Pot nikoli ne pelje ven iz zemljevida. Začetna točka nikoli ni na oviri.

Klic `mozna_pot(0, 0, ">>>>>v>vv>>vvv<v<<^<<<<<", ovire)` preveri pot, označeno na gornjem zemljevidu, in vrne `True`, saj je takšna pot možna.

2. Poenostavitve

Napiši funkcijo `zdruzljivi(ovira1, ovira2)`, ki vrne `True`, če sta oviri (podani s četverkama) združljivi in `False`, če nista. Oviri sta združljivi, če stojita ena ob drugi tako, da se ujemata s celo stranico. Na gornji sliki sta združljivi 15 in 14 (ali 14 in 15), 5 in 6, 11 in 12, 11 in 13. **Niso** pa združljive 15 in 16 ali 2 in 3, 4 in 8, 7 in 5 ...

Napiši funkcijo `zdruzi(ovira1, ovira2)`, ki prejme dve združljivi oviri in vrne oviro, ki jo dobimo, če združimo podani oviri v eno. Če ji podamo, recimo oviri 14 in 15 (klic: `zdruzi((2, 3, 5, 4), (2, 1, 5, 3))`, vrne `(2, 1, 5, 4)`.



Napiši funkcijo `poenostavi(ovire)`, ki prejme seznam ovir. Funkcija ne vrne ničesar, pač pa spremeni podani seznam tako, da so zaporedne ovire, ki jih je mogoče združiti, združene. Pri tem mora ohraniti vrstni red: združena ovira mora biti na tistem mestu v seznamu, kjer sta bili prej posamični oviri. Seznam, ki predstavlja ovire z gornje slike, spremeni v seznam na spodnji sliki.

Predpostaviti smeš, da je seznam urejen tako, da boš vedno združeval oviri na sosednjih mestih v seznamu. (Pazi: 14 in 15 se združita v novo oviro, ki se jima nato pridruži še 16 – tako dobimo oviro 9 na drugi sliki. Ko se združita 11 in 12, pa se jima 13 ne more pridružiti – na drugi sliki imamo zato 7 in 8.)

3. Kisel dež

Izjemno lokalne plohe kislega dežja padajo na posamične kvadratke. Če na isto oviro padejo tri plohe (ali več), je uničena. Pri tem ni pomembno, ali vse tri plohe padejo na isto mesto na oviri ali ne.

Napiši funkcijo `kisel_dez(ovire, plohe)`, ki prejme množico ovir in seznam koordinat ploh. Koordinate ploh so pari (x, y): plohi `(2, 7)` in `(4, 8)` še zadaneta oviro 0, ploha `(5, 10)` pa jo že zgreši. Funkcija mora vrniti množico ovir, ki so po teh plohah še uporabne – torej množico ovir, na katero so padle manj kot tri plohe (lahko tudi nobena).

4. Brez blata

Ovire so koristne tudi za to, da pešči hodijo po njih namesto po mlakah in blatu. Napiši funkcijo `brez_blata(start, cilj, dovoljene_ovire)`, ki prejme dve oviri (kot terki) in množico ovir, na katere je dovoljeno stopiti. Funkcija vrne `True`, če je s start možno priti na cilj in `False`, če ne. Po ovirah lahko hodimo tudi diagonalno: z ovire 4 je možno stopiti na 8.

Da bo možnih več poti, bodo testi za to nalogo dodali oviro `(8, 3, 9, 7)`.

Namig: zadnji argument bo v začetku vseboval vse ovire. Mogoče boš kasneje kakšno oviro sam prepovedal.

OBRNI!

5. Objektno usmerjene ovire

Napiši razred `Ovira`.

- Konstruktor prejme koordinate oglišč (kot posamična števila tipa int, ne kot terko).
- Metoda `ploscina()` vrne ploščino ovire.
- Metoda `ploha(x, y)` obvesti oviro, da pada ploha na koordinato (x, y). Ta morda pripada oviri, morda ne.
- Metoda `uporabna()` vrne `True`, če so na oviro padle manj kot tri plohe.
- Metoda `razdeli_x(x)` vrne oviri (nova objekta vrsta Ovira), ki ju dobimo, če oviro prerežemo pri podani koordinati x. (V bistvu dela natančno nasprotno kot funkcija zdruzi iz prve naloge). Če rez po črti s koordinato x ne prereže ovire, funkcija vrne `None`.
- Metoda `razdeli_y(y)` vrne oviri, ki ju dobimo, če oviro prerežemo pri podani koordinati y. Če rez po tej črti ne prereže ovire, funkcija vrne `None`.