

## Day 4: Camp Cleanup

(besedilo naloge)

Imamo seznam parov intervalov v obliki

```
2-4,6-8
2-3,4-5
5-7,7-9
2-8,3-7
6-6,4-6
2-6,4-8
```

in zanima nas ...

### Prvi del

... koliko je takšnih, pri katerih je eden od njiju v celoti vsebovan v drugem.

Najprej preberemo številke. Lahko delimo niz po - in , lahko se hecemo z regularnimi izrazi, lahko pa preprosto zamenjamo , in delimo po -, pa dobimo štiri številke. Torej

```
pairs = [[int(x) for x in line.replace(",", "-").split("-")]
          for line in open("example.txt")]
```

Vzemimo, recimo, prvi par:

```
x0, x1, y0, y1 = pairs[0]
```

En interval je vsebovan znotraj drugega, če velja

```
x0 <= y0 <= y1 <= x1 or y0 <= x0 <= x1 <= y1
```

False

Srednji primerjavi sta nepotrebni, saj vedno velja  $y0 \leq y1$  in  $x0 \leq x1$ . Vseeno sem ju pisal, saj je alternativa

```
x0 <= y0 and y1 <= x1 or y0 <= x0 and x1 <= y1
```

False

manj pregledna, če dodamo oklepaje,

```
(x0 <= y0 and y1 <= x1) or (y0 <= x0 and x1 <= y1)
```

False

pa je videti že, kot da je zapleteno. Prva različica najbolj jasno pokaže, da gre za takšno in takšno zaporedje števil.

Da rešimo nalogo, moramo le prešteti, kolikokrat je izraz resničen. Ker je `True` enak 1 in `False` enak 0, lahko preprosto seštejemo vrednost tega izraza za vse pare.

```
print(sum(x0 <= y0 <= y1 <= x1 or y0 <= x0 <= x1 <= y1 for x0, x1, y0, y1 in pairs))  
2
```

## Drugi del

... koliko intervalov vsebuje kak skupni element.

Da imata intervala skupni element, mora biti začetek prvega znotraj drugega ali pa začetek drugega znotraj prvega.

```
print(sum(x0 <= y0 <= x1 or y0 <= x0 <= y1 for x0, x1, y0, y1 in pairs))  
4
```