

resitev

January 28, 2024

0.1 Rešitev

Imamo torej

```
[1]: import re
import numpy as np

def anyrange(a, b):
    return np.arange(a, b + 1) if b >= a else np.arange(a, b - 1, -1)

lines = np.array([[int(x) for x in re.findall(r"\d+", v)]
                  for v in open("example.txt")])
```

Kako veliko matriko si moramo pripraviti? Poiskati moramo največjo koordinato x in največjo koordinato y . x -i so v ničtem in drugem stolpcu, y prvem in tretjem.

```
[2]: lines
```

```
[2]: array([[0, 9, 5, 9],
           [8, 0, 0, 8],
           [9, 4, 3, 4],
           [2, 2, 2, 1],
           [7, 0, 7, 4],
           [6, 4, 2, 0],
           [0, 9, 2, 9],
           [3, 4, 1, 4],
           [0, 0, 8, 8],
           [5, 5, 8, 2]])
```

Potrebujemo maksimum ničtega in drugega skupaj ter maksimum prvega in tretjega. Seveda bi to lahko poračunali tako, da bi izračunali

```
[3]: np.max(lines, axis=0)
```

```
[3]: array([9, 9, 8, 9])
```

in to potem združevali (kar je glede na te konkretne številke sicer dolgočasno, vendar razmišljajmo o splošnem primeru). Vendar je, glede na to, da smo slučajno izvedeli za **reshape**, bolj zabavno združiti stolpca.

```
[4]: lines.reshape((-1, 2))
```

```
[4]: array([[0, 9],
           [5, 9],
           [8, 0],
           [0, 8],
           [9, 4],
           [3, 4],
           [2, 2],
           [2, 1],
           [7, 0],
           [7, 4],
           [6, 4],
           [2, 0],
           [0, 9],
           [2, 9],
           [3, 4],
           [1, 4],
           [0, 0],
           [8, 8],
           [5, 5],
           [8, 2]])
```

Naša maksimuma sta potem

```
[5]: np.max(lines.reshape((-1, 2)), axis=0)
```

```
[5]: array([9, 9])
```

Matrika, ki jo potrebujemo, mora biti za 1 večja, ker štejemo od 0.

```
[6]: field = np.zeros(np.max(lines.reshape((-1, 2)) + 1, axis=0))
```

Prvi del naloge zahteva, da upoštevamo le črte, ki so navpične ali vodoravne. Za vsako povečamo ustrezne elemente tabele za 1. Indekse “ustreznih elementov” nam prijazno pove `anyrange`.

```
[7]: for x1, y1, x2, y2 in lines:
      if x1 == x2 or y1 == y2:
          field[anyrange(x1, x2), anyrange(y1, y2)] += 1
```

Kako prešteti neničelne elemente? Takole.

```
[8]: field > 1
```

```
[8]: array([[False, False, False, False, False, False, False, False, False,
           True],
           [False, False, False, False, False, False, False, False, False,
           True],
```

```
[False, False, False, False, False, False, False, False, False,
 True],
[False, False, False, False,  True, False, False, False, False,
 False],
[False, False, False, False, False, False, False, False, False,
 False],
[False, False, False, False, False, False, False, False, False,
 False],
[False, False, False, False, False, False, False, False, False,
 False],
[False, False, False, False,  True, False, False, False, False,
 False],
[False, False, False, False, False, False, False, False, False,
 False],
[False, False, False, False, False, False, False, False, False,
 False]])
```

Le tole moramo sešteti.

```
[9]: np.sum(field > 1)
```

```
[9]: 5
```

Za drugi del pa le izpustimo if in bomo dobili še diagonale.

Lahko pa rešimo oba dela hkrati. Celotna rešitev obeh delov je:

```
[10]: import re
import numpy as np

def anyrange(a, b):
    return np.arange(a, b + 1) if b >= a else np.arange(a, b - 1, -1)

lines = np.array([[int(x) for x in re.findall(r"\d+", v)]
                  for v in open("example.txt")])

for part in (0, 1):
    field = np.zeros(np.max(lines.reshape((-1, 2)) + 1, axis=0))
    for x1, y1, x2, y2 in lines:
        if part == 1 or x1 == x2 or y1 == y2:
            field[anyrange(x1, x2), anyrange(y1, y2)] += 1
    print(np.sum(field > 1))
```

```
5
```

```
12
```