PYTHON

Osnove 7

Python urejevalnik se nahaja na strani:
`https://www.w3resource.com/python-exercises/python-basic-exercises.php#EDITOR`
Naloge Kattis:`https://github.com/minidomo/Kattis`

1. **Permutacije** Izpiši vse mogoče razporedbe elementov seznama `{s = [i for i in range(1,n+1)]}`. Vzemimo, da je $n < 6$. **Vhod:** Iz standardnega vhoda preberi naravno število n.
   Izhod: Na standardni izhod izpiši seznam permutacij elementov seznama `{s}`.
   Primer:

   ```
   Vhod:
   3
   Izhod:
   [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
   ```

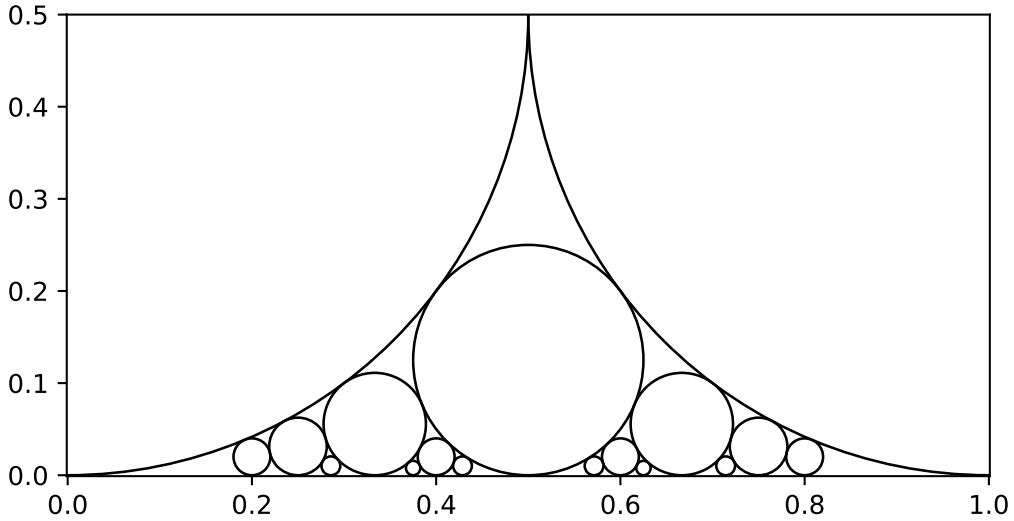
3. **Fareyevi ulomki**
   Ulomke bomo zapisali v obliki dvojic. Dvojica predstavlja (p, q) predstavlja ulomek p/q. Začeli bomo z dvema ulomkoma $[(0, 1), (1, 1)]$. Seznam bomo širili tako da bomo med dva ulomka (p_1, q_1) in p_2, q_2, za katera velja, da je $(p_1, q_1) < (p_2, q_2)$ vrinili ulomek $(p_1 + p_2, q_1 + q_2)$. Pokaži, da velja

   $$(p_1, q_1) < (p_1 + p_2, q_1 + q_2) < (p_2 + q_2).$$

   Vhod: Iz standardnega vhoda preberi naravno število n.
   Izhod: Na standardni izhod izpiši seznam ulomkov po n-ti iteraciji.
   Primer:

   ```
   Vhod:
   3
   Izhod:
   Začnemo z [(0,1),(1,1)]
   1. [(0,1),(1,2),(1,1)]
   2. [(0,1),(1,3),(1,2),(2,3),(1,1)]
   3. [(0,1),(1,4),(1,3),(2,5),(1,2),(3,5),(2,3),(3,4),(1,1)]
   ```

4. **Fareyeve krožnice** Krožnice, ki imajo središča v Fareyevih ulomkih $(\frac{p}{q}, \frac{1}{2q^2})$ in polmer $r = \frac{1}{2q^2}$, se med seboj dotikajo. Če sta abscisi dveh dotikajočih se krožnic enaki $x_1 = p_1/p_2$ in $x_2 = p_2/q_2$, potem je abscisa tretje krožnice, ki se dotika obeh enaka $(p_1 + p_2)/(q_1 + q_2)$, ki pa je spet Fareyeva krožnica.



Uporabi program za risanje:

```
#!/usr/bin/env python3
import matplotlib.pyplot as plt
from farey import farey

x_ = [-0.001, 1.001]
y_ = [-0.001, 1/2]
fig, ax = plt.subplots()
ax.set(xlim=(x_[0],x_[1]), ylim=(y_[0],y_[1]),
    autoscale_on=False, clip_on=True, aspect='equal')

def circle_(x, y, r, fill, color):
    return plt.Circle((x, y), r, color=color, clip_on=True, fill=fill)

if __name__ == '__main__':
    n = int(input('n -> '))
    for p, q in farey(n):
     cc = circle_(p/q, 1/2/q**2, 1/2/q**2, False, 'black')
     ax.add_patch(cc)
    plt.show()
```

Rešitve 7

Permutacije

```python
#!/usr/bin/env python3
import random

def permute(s):
    if len(s) == 1:
        return [s]
    else:
        res = []
        for i in range(len(s)):
            sub_permute = permute(s[:i] + s[i+1:])
            for x in sub_permute:
                res.append([s[i]] + x)
    return res

def permute_cycles(s):
    ds = dict(enumerate(s))
    cycles = []
    while ds:
        si = iter(ds)
        cycle = []
        it = next(si)
        while it in ds:
            cycle.append(ds[it])
            jt = ds[it]
            del ds[it]
            it = jt
        cycles.append(cycle)
    return cycles

if __name__ == '__main__':
    perm = list(range(1, 11))
    random.shuffle(perm)
    print(perm, permute_cycles(perm))
```

Fareyevi ulomki

```python
#!/usr/bin/env python3

def plus(a, b):
    return (a[0] + b[0], a[1] + b[1])

def farey(n):
    res = [(0,1),(1,1)]
    for _ in range(n):
        j = 1
        for i in range(1,len(res)):
            res.insert(j,plus(res[j-1],res[j]))
            j = j + 2
    return res

if __name__ == '__main__':
    n = int(input('n -> '))
    print(farey(n))
```