

Nanotori

Tomas Rode
Enej Kovač

6. november 2019

1 Navodila

Nanotor je 3-regularen graf na torusu. Vsak nanotor lahko dobimo tako, da na šeskotni mreži enačimo nasprotni stranici danega paralelograma. Torej je vsak nanotorus določen z dvema vektorjema v ravnini, (k, l) in (m, n) , za katera velja $k^2 + l^2 \neq 0$ in $m^2 + n^2 \neq 0$. Projekt je sestavljen iz štirih podnalog:

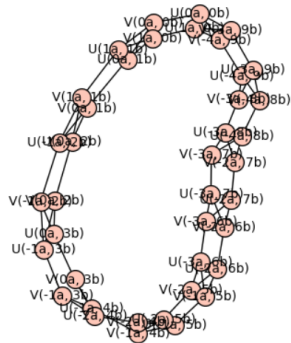
1. V prvem delu naloge ustvarite funkcijo, ki v *Sage* konstruira nanotor, za dane k, l, m in n .
2. S pomočjo funkcij v *Sage* preučite nekaj lastnosti nanotorov: za dan (k, l, m, n) določite število vozlišč, premer, tranzitivnost, ...
3. Za $v \in V(T)$, poljubno vozlišče nanotora, določite število vozlišč na razdaljah $i : 1 \leq i \leq \text{diam}(T)$. Poiščite formulo za dane i, k, l, m, n .
4. Naj bo T nanotorus tipa (k, l, m, n) . Ugotovite ali obstaja nanotorus tipa $(k', 0, m', n')$, izomorfen T . Če obstaja, ugotovite odnos med (k, l, m, n) in (k', m', n') .

2 Opravljeno delo (do 5. 11. 2019)

S pomočjo objektnega programiranja sva v *Sage* zapisala funkcijo, ki konstruira nanotor s k, l, m in n . Na sponjih slikah lahko vidimo nekaj primerov.

```
G = nanotorus(2,0,-5,10)
print(G.is_vertex_transitive(), G.order(), G.size(), G.degree().count(3) == len(G.degree()), G.diameter())
#(tranzitivnost, št. vozlišč, št. povezav, vse točke reda 3, premer)
G.plot()

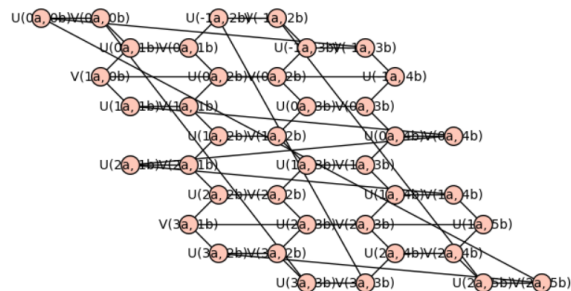
(True, 40, 60, True, 10)
```



Na prvi sliki sva dovolila, da *Sage* prerazporedi točke po prosotru tako, da se čim bolje vidi oblika grafa. Pri tem se nekoliko izgubi šestkotna mreža, na kateri smo graf ustvarili. Pri vsakem grafu sva preverila še tranzitivnost, število vozlišč in povezav, red vozlišč in premer grafa.

```
G = nanotorus(4,2,-2,4)
print(G.is_vertex_transitive(), G.order(), G.size(), G.degree().count(3) == len(G.degree()), G.diameter())
#(tranzitivnost, št. vozlišč, št. povezav, vse točke reda 3, premer)
G.plot()

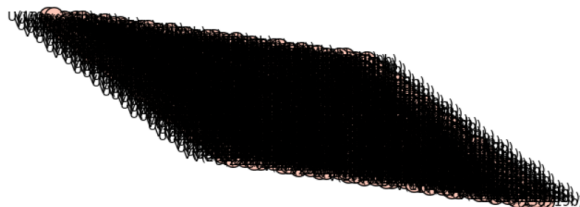
(True, 40, 60, True, 6)
```



Pri drugem grafu sva ohranila koordinate vozlišč na šestkotni mreži. Izmed vseh grafov je torej tukaj izvorna šestkotna mreža najbolj razvidna.

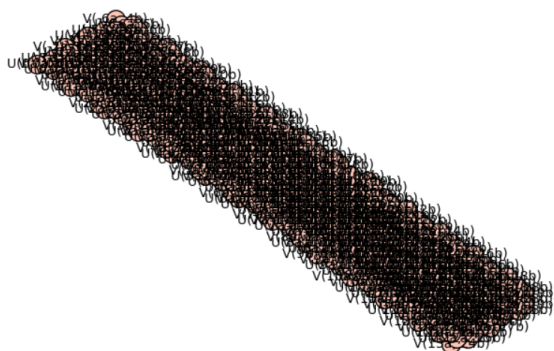
```
G = nanotorus(10,-35,13,20)
print(G.is_vertex_transitive(), G.order(), G.size(), G.degree().count(3) == len(G.degree()), G.diameter())
#(tranzitivnost, št. vozlišč, št. povezav, vse točke reda 3, premer)
G.plot()

(True, 1310, 1965, True, 35)
```



```
G = nanotorus(-7,5,13,25)
print(G.is_vertex_transitive(), G.order(), G.size(), G.degree().count(3) == len(G.degree()), G.diameter())
#(tranzitivnost, št. vozlišč, št. povezav, vse točke reda 3, premer)
G.plot()

(True, 480, 720, True, 36)
```



Tretji in četrti graf imata veliko vozlišč, kar ju naredi zelo nepregledna, omogoči pa, da preverimo omenjene lastnosti nanotorov tudi za velike grafe.

3 Načrt za nadaljnje delo

Delo bova nadaljevala pri naslednjih točkah iz navodil, torej bova najprej ugotavljala lastnosti nanotorov. Iz teh ugotovitev bova potem preverjala, kakšne so razdalje v nanotoru in kdaj sta nanotora izomorfna.