

Kodutöö 1, Laiutiotsing

Urmas Pitsi, 8.sep.2019

Käesoleva töö tulemuseks on Pythonis kirjutatud lühima tee leidja kasutades laiutiotsingu algoritmi. Sisendiks on ASCII-art formaadis tekst, milles määratud algus (s) ja lõpp-punkt (D) ning lubatud/keelatud käigukohad (tühik/tärn vastavalt). Lahenduse visualiseerimiseks lisasin vastava funktsiooni, mis kuvab tulemusena leitud lühimat teekonda sisendi peal (vt Joonis 1). Lisaks täiendasin algoritmi nii, et soovi korral saaks lähimaid naabreid läbida juhuslikus järjekorras. See võimaldab leida alternatiivseid, kuid samaväärseid lühimaid teekondasid alguspunktist lõpp-punkti.

Lahendus

Ülesandes antud kahele sisendile leidsin järgnevad vastused:

“lava_map1”: lühim tee 18 sammu alguspunktist lõpp-punkti.

“lava_map2”: lühim tee 30 sammu alguspunktist lõpp-punkti.

Laiutiotsingul hakkame järjest läbi käima lähimaid naabreid, alustades alguspunkti ümbrusest: 1-vasakul, 1-üleval, 1-paremal. Kontrollime kas on võimalik liikuda antud koordinaadile (tühik) ja kas me pole seda enne juba läbinud. Kui jah, siis lisame selle punkti ootejärjekorda, et omakorda tema naabreid kontrollida. Nii teeme, kuni oleme leidnud lõpp-punkti (“D”).

Koodinäidis

```
rl = Rajaleidja(lava_map1, (start_row, start_col))
shortest_path = rl.shortest_path_breadth_first(randomize_order=False)
rl.draw_shortest_path(randomize_order=False)
```

Joonis 1

“lava_map1” lühim tee	“lava_map2” lühim tee
[' ** ** ',	[' ***** ',
' *** D.... *** ',	' ***** D.... ***** ',
' *** ',	' ***** ',
' . ***** ',	' ***** ***** ',
' **** . ***** ',	' ***** ***** ',
' *** . ***** ',	' ***** ***** ',
' ** ***** ',	' ***** ***** ',
'***** . ***** *** ',	' ***** ***** ',
'***** . ** ',	' ***** ***** ',
'*** ',	' *** ***** ',
' ** . ***** ',	' * ***** ***** ',
'** ***** ',	' ***** ***** ',
'*** ***** ',	' *** ***** ',
' . ',	' . ',
' ** . ***** ',	' . ',
'*** ***** ',	' . ',
' . ',	' . ',
' s ',	' s ',