Algoritmi
Študijsko leto 2021/2022

3. domača naloga

Poročilo domače naloge

Ivo Pajer
Vpisna št. 63180218

Novo mesto, 2. maj 2022

Kazalo

1 Problem 1 - Konveksna ovojnica

1.1 A - Algoritem za izračun unije ovojnic

Naša naloga je, da napišemo algoritem, ki sprejme 2 konveksni ovojnici in vrne konveksno množico unije vhodnih konveksnih ovojnic. Za unijo bomo uporabili metodo Rotating Calipers [1], ki nam omogoči, da se naš algoritem zbliža na časovno zahtevnost $\mathcal{O}(n)$. Glavna ideja je, da začnemo na najbolj levi zgornji točki, označimo jo T1. in se nato pomikamo desno, dokler ne zadanemo točke, ki mora biti vsebovana v konveksni ovojnici. Od vsake točke pogledamo, katera točka je najdlje in jo dodamo v unijo. Ko je iskalni kot večji od π, se pomaknemo naprej v naslednjo točko T1, ki še ni v novi konveksni ovojnici.

1.2 B - Sekanje konveksnih ovojnic

Naša naslednja naloga je, da naš program vrne, ali se konveksni ovojnici sekata, zato moramo najprej določiti, kdaj se ovojnici sekata. Privzeli bomo, če so ovojnice vsebovanje ena v drugi se ne bodo sekali. Sekali se bota, če se vse črte iz konveksne ovojnice sekata ali dotikata ena z drugo. Zato bo naš algoritem enostavno primerjal vse črte iz konveksne ovojnice. Če se bo kakšna črta sekala ali dotikala, bo naš algoritem vrnil, da se ovojnici sekata.Če do tega ne bo prišlo in bomo pogledali že vse črte med seboj, bomo vrnili, da se ovojnici ne sekata.

2 Problem 2 - Delaunay triangulacija

3 Problem 3 - Drevesa

3.1 A - Primerjava drevesnih struktur

Naša nalogaje primerjati kd-trevesa, četrtinska drevesa in intervalna drevesa. Vsaka ima distinktno prednost pred drugimi. Vsa ta drevesa so namenjena iskanju točk v ravnini.

- **kd-drevesa** - Glavna prednost pred drugimi je, da imajo manjšo prostorsko zahtevnost $\mathcal{O}(n)$, kot četrtinska drevesa, ki imajo prostorsko

zahtevnost $\mathcal{O}(n)$ v najboljšem primeru, saj ima vsaka točka v drevesu 4 otroke. kd-drevesa imajo tudi boljšo prostorsko zahtevnost kot intervalna drevesa, ki imajo prostorsko zahtevnost $\mathcal{O}(n\ log^{d-1}\ n)$, kjer je d dimenzija točke (v našem primeru 2).

- **četrtinska drevesa** - Glavna prednost tega drevesa je, da je vstavljanje v drevo hitrejše in konsistentno, saj vrstni red dodajanja tčk v drevo ni važno, kot je pri drugih dveh.
- **intervalna drevesa** - Glavna prednost te streukture je, da omogoča hitre poizvedbe $\mathcal{O}(log^d n)$ v primerjavi z kd-drevesi $\mathcal{O}(n)$, v primerjavi z četrtniskim drevesom pa ima hitrejše poizvedbe, ko so podatki manj ugodno razporejeni primer: vsi naraščujoče med seboj.

3.2 B - Osmiška drevesa

Naša naloga v tem delu je opisati algoritem za izdelovanje osmiškega drevesa. Osmiško drevo je zelo podobno četrtinskemu drevesu, le da je namenjen delu z 3d podatki namesto 2d podatki. Najprej določimo smeri v katere bomo dodajali elemente osmiškega drevesa (npr. glede na binarno število - 0 padajoče v tej smeri, 1 naraščujoče v tej smeri). Nato moramo določiti meje, v katerih se bodo naše točke nahajale (največje in najmanjše vrednosti točk). Določimo še našo vhodno točko in prostor razdelimo na 8 enako velikih delov, tako, da so točke vsebovanje v pod-drevesu večje ali manjše od vrednosti delitve. Ta proces delitve ponavljamo, dokler nismo zadovoljni z natančnostjo našega drevesa in v pravi kradrant umestimo željeno točko.

Literatura

[1] "Wikipedia: Rotating calipers," Dec 2020.