

Seminarski Rad

Geometrija

Konstrukcija konveksnog omotača – MergeHull Algoritam

Dokumentacija za program

Aleksandra Mačkić

Nikola Grujić

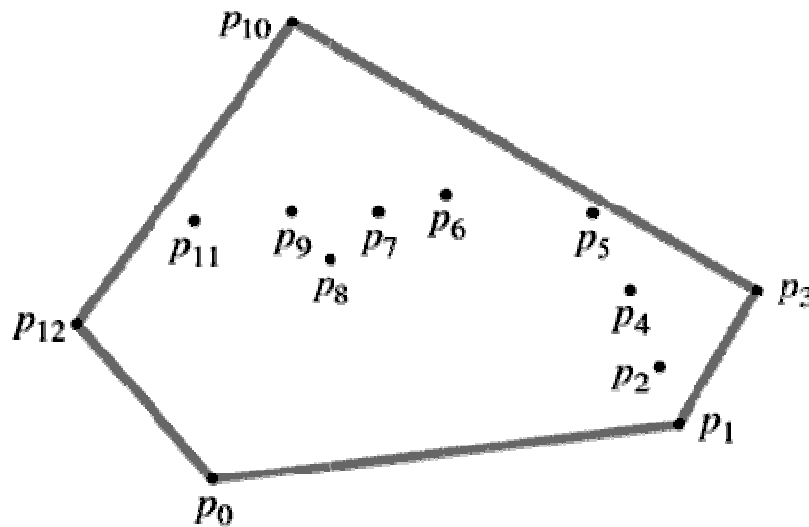
Nenad Avramović

Stefan Đorđević

Šta je konveksni omotač?

Konveksni omotač (eng. convex hull) skupa tačaka Q je najmanji konveksni poligon P takav da svaka tačka iz skupa Q leži u unutrašnjosti poligona ili na njegovoj granici. Drugačije gledano ako zamislimo da svaka tačka predstavlja ekser koji viri iz daske, onda bi konveksni omotač bila figura koju formira zategnuta gumica koja obuhvata sve eksere.

Skup tačaka i njihov konveksni omotač:



Određivanje konveksnog omotača skupa tačaka je interesantan problem sam po sebi. Međutim, algoritmi za neke druge probleme kompjuterske geometrije počinju određivanjem konveksnog omotača. Posmatrajno, na primer, dvodimenzioni problem određivanja para najudaljenijih tačaka: za dati skup od n tačaka u ravni potrebno je odrediti dve tačke koje su međusobno najviše udaljene. Može se dokazati da te dve tačke pripadaju konveksnom omotaču. Pored ovoga algoritam za konveksni omotač našao je primenu u robotici, obradi digitalnih slika i prepoznavanju oblika.

MergeHull algoritam

MergeHull algoritam spada u grupu algoritama za konstrukciju konveksnog omotača implementiranih algoritamskom strategijom „podeli pa vladaj“ (eng. divide and conquer). Ideja ovog algoritma je identična ideji **MergeSort** algoritma.

Za dati skup S od n tačaka u ravni, prvo se vrši podela skupa na dva podskupa S_1 i S_2 približno jednakih dimenzija, a onda se odvojeno konstruiše konveksni omotač za svaki od skupova S_1 i S_2 . Na kraju se omotači spajaju u omotač koji odgovara početnom skupu

tačaka. Razdvajanje skupa tačaka na podskupove se vrši na osnovu vrednosti X koordinate tačaka, podskup S_1 se formira od tačaka sa manjom X koordinatom, a podskup S_2 od onih tačaka sa većom X koordinatom, a pošto je ovo rekurzivna procedura, uslov rekurzivnog koraka je da je skup veći od 3 tačke. Ova podela se implementira tako što se sve tačke sortiraju po X , a zatim se u linearnoj vremenskoj složenosti pronalazi srednja vrednost. Kad se ispuni uslov rekurzije, može se "ručno" konstruisati omotač za 2(3) tačke. Problem je kako spojiti te omotače u jedinstveni.

Ideja spajanja je u tome što se pronađu 4 tačke (dve u levom, dve u desnom delu) takve da kad se one spoje, ostale tačke su ispod (ako je u pitanju gornja spojnica) ili iznad (donja spojnica). Te dve "spojnice" pripadaju skupu linija konveksnog omotaca, i još je potrebno obrisati unutrašnje linije početnih omotača.

Implementacija

Skup tačaka je implementiran kao niz struktura `Niz` koja je definisana u zaglavlju `funkc.h`, čije su komponente struktura `Tacka`, inače definisana u zaglavlju `crtanje.h`, i dve celobrojne vrednosti `sled` i `pred` koje će biti upotrebljene za povezivanje tačaka. Na početku se nasumično generisan skup tačaka iz tekstualne datoteke `koord.txt` učitava u niz. Potom se niz sortira upotrebom implementiranog `qsort` algoritma. Sortiranje se obavlja tako što funkcija `cmp(argument funkcije qsort)` prvo poredi X koordinatu svake dve tačke, pa u slučaju da su X koordinate jednake, kao manju uzima onu sa manjom Y koordinatom. Potom se za tako sortiran niz tačaka poziva funkcija `Merge_hull`, koja kao argumente pored niza dobija još i indekse početnog i krajnjeg člana.

Funkcija `Merge_hull` implementirana je tako da se niz deli na dva po veličini jednaka niza a zatim se funkcija `Merge_hull` rekurzivno poziva za svaki od njih. To se radi sve dok se ne dođe u situaciju da se funkcija pozove za dve ili tri tačke. U slučaju poziva za dve tačke spajanje je trivijalno i vrši se tako što svaka tačka za vrednost komponenta `pred` i `sled` dobija vrednost indeksa druge tačke (slično pokazivačima u implementaciji lista). U slučaju poziva za tri tačke mora se voditi računa o tome kojim redom se tačke spajaju. Za to nam služi funkcija orijentacija koja kao argumente dobija pokazivač na početak niza i početni indeks. Ona u stvari računa orijentaciju trougla koji se formira sa te tri tačke, i u odnosu na rezultat zaključujemo kojim redom povezujemo te tačke.

Na kraju svih rekurzivnih poziva koji su kreirani, izmenjen je sadržaj niza tako što su promenjene vrednosti komponenta `pred` i `sled` tako da su elementi konveksnog omotača povezani tim vrednostima.

Main funkcija:

Posle deklaracije i inicijalizacije promenljivih iz fajla `koord.txt` učitavaju se koordinate u dinamički alociran niz. Vrednosti `pred` i `sled` komponenta nisu bitne jer će se one promeniti u validne, ako bude potrebe za neku tačku da promenimo tu vrednost.

Qsort() se poziva kao sistemska funkcija koja sortira upotrebom prethodno opisane funkcije cmp.

Iscrtavamo sve tačke iz koord.txt, i pozivamo merge hull funkciju za ceo niz. Po završetku poziva iscrtavaju se duži koje su ustvari konveksni omotač.

Upotreba programa:

Pokrenuti generator.exe i uneti broj tacaka koje zelimo generisati(broj izmedju 2 i 100, samo da bi slika bila dovoljno jasna, iako algoritam radi za proizvoljan broj tačaka).

/*u svrhe testiranja algoritam je bio primenjen na 20.000 objekata*/

Posle toga pokrenuti program MergeHull.exe koji ce pronaci konveksni omotač za zadate tačke i iscrtati sliku vezanu za dati problem. Tacke je moguće zadati i rucno tako sto se u datoteku koord.txt upišu zeljene koordinate td. u svakom redu budu 2 realna broja koja predstavljaju x i y koordinatu tacke.

/*dopuna: U glavni program je dodata i mogućnost crtanja 3 test primera(čiča gliša, šaka i auto.), pratiti uputstva komande linije po pokretanju programa*/