Projekat iz predmeta
Uvod u kompijtersku geometriju

Berin Spahović

05/22/19

1.1. Definicija projekta

Dato je n tačaka u ravni (n je neparan broj) tako da nikoje tri nisu na istoj pravoj i nikoje 4 nisu na istoj kružnici. Odrediti kružnicu koja prolazi kroz tri od datih n tačaka, tako da od preostalih $n - 3$ tačaka, jednak broj tačaka je unutar kruga kao i van kruga.

1.2. Ideja rjesešenja problema

Glavna ideja za rješenje ovog problema je u samoj osobini kružnice. Naime poznato je da je periferijski ugao nad prečnikom uvijek fiksan i iznosi $90°$. Ali isto tako, sve tačke koje se nalaze unutar kružnice grade tup ugao sa prečnikom, dok tačke van njega grade oštar.
Uzmimo neku prozivoljnu kruznicu k i proizvoljnu tačku X van kruga k. Ugao koji gradi prečnik kruga k sa tačkom X je manji od svih uglova unutar te kružnice. Sada ako povučemo kružnicu koja spaja dva vrha prečnika i tačku X dobijamo kružicu gdje sve tačke unutar nje grade ugao manji od ugla kod tačke X. A sve tačke na toj kružici zaklapaju isti ugao sa prečnikom kao i X. (Osobina tetivnosti).
Ova intuicija nas vodi ka rješenju našeg problema.

1.3. Opis algoritma

Neka je S skup tačaka i neka je K konveksni omotač datog skupa S.

Algoritam:

- Odabrati duž koja pripada konveksnom omotaču skupa tačaka
- Ta duž će predstavljati prečnik naše tražene kružnice.
- Odrediti ugao koji gradi ovaj prečnik sa svim tačkama iz S.
- Odrediti $\frac{n-1}{2}$ po veličini ugao i sačuvati tu tačku.
- Ta tačka predstvalja treću tačku naše kružnice i ona ima osobinu da polovi broj tačaka na pola.

Dokažimo da je baš ta tačka naša tražena i da ova kružnica polovi broj tačaka na pola:
Neka je AB duž konveksnog mnogougla. Pošto ona pripada konveksnom omotaču ova duž ima osobinu da prava koja prolazi kroz nju dijeli ravan na dvije poluravni od koje jedna nema niti jednu tačku. Uzmimo sada tačku M koja gradi največi ugao sa duži AB. Nacrtajmo kružicu koja spaja tačke A, B i M. Predpostavimo da postoji neka tačka X iz S unutar ove kružnice. Iz osobine kružnice imamo $\angle AXB > \angle AMB$ što je kontradikcija, budu[ci da je M tačka koja gradi maksimalni ugao. Dalje, svi ovi uglovi su u intervalu $(0°, 180°)$ jer ne postoje tri tačke koje su kolinearne. Isto tako, svi ovi uglovi sa duži AB su različiti jer ne postoje četri tačke koje se nalaze na istoj kružnici.
Uzimajući sljedeću tačku sa najvećim uglom, zaključujemo da tačka M pripada unutar te kružnice i nikoja druga. Ovo nas dovodi do zaključka da ako uzmemo $k-$ti najveći ugao, unutar te kružnice se nalazi tačno $k-1$ tačaka.
Neka je $n = 2k + 3$. Nako što smo uzeli dvije tačke iz konveksnog omotača ostaje nam još $2k + 1$. Od ovih tačaka trebamo naći tačku koja ima jednak broj tačaka unutar i van kružnice. To je $k + 1$ po redu tačka tj:

$$k + 1 = \frac{n-1}{2}$$

1.4 Implementacija algoritma

Za ovaj algoritam koristio sam sljedeće strukture:

- $struct\ Tacka(int\ x, int\ y)$: Struktura koja čuva kordinate tačaka.
- $struct\ Duz(Tacka\ pocetna, Tacka\ krajnja)$:
- $struct\ Trougao(Tacka\ A, Tacka\ B, Tacka\ C)$: Ovdje smo strukturu trougao čuvali preko tri nekolinearne tačke.
- $struct\ Prava(double\ a, double\ b, double\ c)$: Jednačina prave je:

$$ax + by = c$$

- $struct\ Krug(Tacka\ centar, double\ radius)$: Svaki krug je definisan preko centra i dužine njegovog poluprečnika.

Sljedeće metode su bile potrebne za rješavanje datog problema:

- $Prava\ dajPravu(Tacka, Tacka)$: Metoda koja vraća pravu koja prolazi kroz zadate dvije tacčke.
- $Krug\ vratiKrug(Tacka, Tacka, Tacka)$: Vratiti kružnicu koja prolazi kroz tri nekolinearne tačke.
- $Prava\ vratiSimetraluDuzi(Duz)$: Simetrala duži je prava koja polovi zadatu duž i okomita je na nju.
- $Tacka\ vratiPresjekPravi(Prava, Prava)$: Tacka presjeka dvije prave.
- $double\ vratiUdaljenost(Tacka, Tacka)$: Euklidska udaljenost izmedju dvije tačke.
- $double\ vratiUgao(Tacka, Tacka, Tacka)$: Vratiti ugao koji gradi treća tačka sa prvim dvjema. Ovdje smo koristi kosinusnu teoremu koja vrača ugao iz intervala $(0°, 180°)$.

1.5 Vremenska kompleknost

Pronači dvije tačke konveksnog omotača je moguće uraditi u $O(n)$ vremenu. Naime, prva tačku predstavlja najlijevlju tačku po x-osi. Drugu tačku tražimo tako što odredimo tačku koja gradi najmanji ugao sa pravom koja je paralelna sa x-osom i prolazi kroz prvu tačku.
Nakon ovog djela, trebamo odrediti $\frac{n-1}{2}$ po veličini ugao.
Jedan od naivnih načina je sortirati sve uglove, što će uzeti $O(nlogn)$.
Poboljšanje ove kompleksnosti je korsiti $minHeap$ sa fiksnim vremenom izvršavanja: $O(n + klong)$.
Ono što sam ja koristio je $QuickSort$ algoritam sa prosječnim vremenom izvršavanja $O(n)$, sa worst-casom $O(n^2)$ gdje scenarijo nastupa kada uvijek uzimamo največi ili najmanji element za pivot, tj. kad je niz već sortiran. Rekurzivana relacija za worst-case je:

$$T(n) = T(n-1) + \theta(n)$$

Best-case nastupa kada za pivot uvijek uzimamo srednji element. Sljedeća rekurzivna relacija daje opis ovog slučaja:

$$T(n) = 2T(n/2) + \theta(n)$$

1.6 Dodatne specifikacije

Ubacio sam mogućnost interakcije sa userom. Naime, korisnik je u mogućnosti unijeti proizvoljan broj tačaka gdje ima mogućnost random generisanja tačaka po prostoru. Ono što treba popraviti u ovom generisanju, jeste da ne mogu nastupiti gore navedena dva uslova pa iz tog razloga je moguče da generator generiše dosta tačaka koje su kolinearne.
Takodjer, korisnik ima mogućnost ručnog dodavanja tačaka u prostoru.