

# Dokumentacija za Projekat UKG

## Rijad Mutapcic

### Tema 1:

Neka je dat skup od  $n$  kružnica u ravni. Potrebno je ispitati da li među tim kružnicama postoje dvije koje se sijeku (ako je jedna kružnica unutar druge, one se ne sijeku). Za rješavanje problema predstaviti svaku kružnicu kao uniju dvije polukružnice (gornje i donje), te onda provjeravati da li se polukružnice sijeku.

a) Dizajnirati i implementirati  $O(n \log n)$  sweep-line algoritam koji rješava dati problem;

b) Objasniti zašto je bilo neophodno svaku kružnicu podijeliti na polukružnice.

### Tehnicke stvari

Prvu stvar što trebamo uraditi su tehnicke prirode. Napravimo funkciju koja klikom na dugme Generisi Krugove generise nasumicno broj kruznic koliko je dato. Nakon toga moramo napraviti funkciju da korisnik doda svoje kruznicе. Ovo uradimo tako da kada je radio button Dodavanje kruga stisnuto prvi klik na canvas odredjujemo centar a drugim klikom odredjujemo poluprecnik date kruznicе.

Sada mozemo na canvas da nacrtamo koliko god zelimo kruznicа ostane nam samo da napravimo funkciju koja provjerava da li se sijeku te kruznicе.

### Algoritam

Ovaj problem da bi se rijesilo u vremenu  $O(n \log n)$  trebamo da koristimo modifikovanu verziju sweep line algoritma. Za sweep line algoritam poznato

je da moramo imati event-ove koje kada sweep line dodje do njih da nesto uradi. U nasem slucaju imamo 4 eventa za svaku kruznicu, a ta 4 eventa su:

1. Kada dodjemo do gornje lijeve tacke kruznice (gornja lijeva tacka je definisana kao Tacka( $C.x - radius, C.y - radius$ ) gdje su  $C.x$  I  $C.y$  tacke centra kruznice I radius je poluprecnik kruznice)
2. Kada dodjemo do gornje desne tacke kruznice (gornja desna tacka je definisana kao Tacka( $C.x + radius, C.y - radius$ ) gdje su  $C.x$  I  $C.y$  tacke centra kruznice I radius je poluprecnik kruznice)
3. Kada dodjemo do donje lijeve tacke kruznice (donja lijeva tacka je definisana kao Tacka( $C.x - radius, C.y + radius$ ) gdje su  $C.x$  I  $C.y$  tacke centra kruznice I radius je poluprecnik kruznice)
4. Kada dodjemo do donje desne tacke kruznice (donja desna tacka je definisana kao Tacka( $C.x + radius, C.y + radius$ ) gdje su  $C.x$  I  $C.y$  tacke centra kruznice I radius je poluprecnik kruznice)

Sada trebamo sve ove event-ove tj tacke sortirati u priority queue. Sortiramo ih po x koordinati I u slucaju da imaju istu x koordinatu onda sortiramo ih po y koordinati.

Nakon sto smo ih sortirali trebamo da napravimo stablo koja ce primati u ovom slucaju polukruznice. Moramo svaku kruznicu podijeliti na polukruzine zato sto se moze desiti da imamo kruznicu A unutar kruzine B pa bi nam u tom slucaju sweep line trebao imati redosljed A gornja, B gornja, B donja, A donja sto da nismo podijelili kruznicu na polukruznice ne bismo imali.

Za stablo moramo imati kriteriji po kojem cemo ubacivati sve ove polukruznice. Posto sam polukruznice tako definisao mozemo da koristimo isti kriteriji kao I za duzi u obicnom sweep line.

Zatim za svaki od 4 tipa eventa imamo jedan if. Za lijeve tacke gornje I donje polukruznice ubacimo ih u stablo I provjeravamo da li je pocetni cvor ili kranji. Ako nisu onda trebamo naci njegov prethodnik I sljedbenik tj to su oni cvorovi u stablu koje su prije tj poslije njega(njegovi susjedni). Zatim za svaki susjednu polukruznicu provjeravamo da li se sjeku kruznice. Ako se sijeku onda break ako ne nastavljamo dalje.

Kada dodjemo do desne tacke gornje I donje polukruznice za njih provjeravamo da nisu pocetni I kranji cvor u stablu pa postoje prethodnik I sljedbenik. Provjeravamo da li se kruznice prethodnika I sljedbenika sijeku ako da onda break ako ne onda izbacimo iz stabla tu polukruznicu I nastavimo dalje.