Problemi najbližih točaka

Višedimenzionalni "podijeli pa vladaj"

Matea Čelar

PMF-MO

22. siječnja 2019.

Sadržaj

Sadržaj

Uvod

- Problemi vezani za skupove točaka u višedimenzionalnom prostoru - N točaka u k-dimenzionalnom prostoru
- Takvi problemi se prirodno javljaju u geometriji za $k = 2, 3$
- Općenito - statistika, analiza podataka, ...

“Podijeli pa vladaj”

N točaka u
k-dimenzionalnom
prostoru

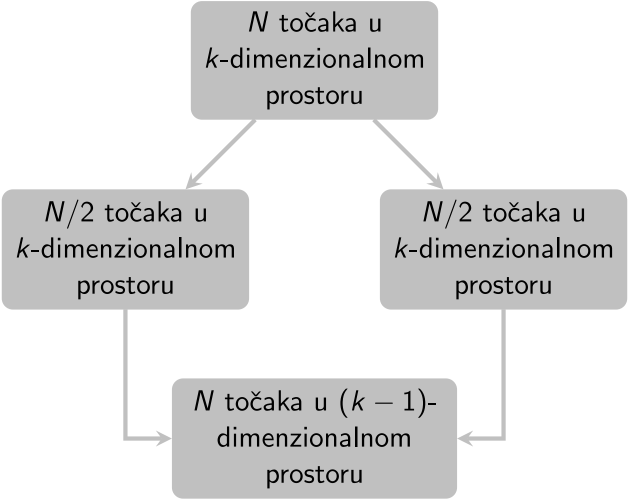
"Podijeli pa vladaj"

N točaka u k-dimenzionalnom prostoru

$N/2$ točaka u k-dimenzionalnom prostoru

$N/2$ točaka u k-dimenzionalnom prostoru

"Podijeli pa vladaj"



“Podijeli pa vladaj”

Da bismo riješili problem za N točaka u k-dimenzionalnom prostoru, najprije rekurzivno riješimo dva problema za $N/2$ točaka u k-dimenzionalnom prostoru, i zatim rekurzivno riješimo još jedan problem za N točaka u $(k-1)$-dimenzionalnom prostoru.

Sadržaj

Problemi najbližih točaka

- Problemi vezani za udaljenost ("blizinu") točaka u prostoru
- Mjera udaljenosti - standardna euklidska metrika u $\mathbb{R}^k$
- Možemo promatrati i druge metrike

Sadržaj

Parovi točaka fiksne udaljenosti

- dan je skup S od N točaka u $\mathbb{R}^k$
- ispisati sve parove točaka iz S čija je udaljenost manja ili jednaka od $d > 0$
- S ima gomilište - složenost $\mathcal{O}(N^2)$
- promatramo *raspršene* skupove:

Parovi točaka fiksne udaljenosti

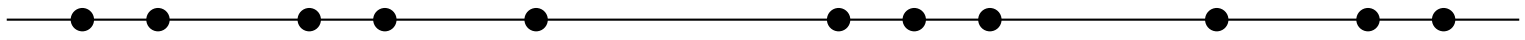
- dan je skup S od N točaka u $\mathbb{R}^k$
- ispisati sve parove točaka iz S čija je udaljenost manja ili jednaka od $d > 0$
- S ima gomilište - složenost $\mathcal{O}(N^2)$
- promatramo *raspršene* skupove:

Definicija 1

Za skup $S \subseteq \mathbb{R}^k$ kažemo da je **raspršen** *ako postoje konstante $C \in \mathbb{N}$ i $d > 0$ takve da nijedna kugla radijusa d ne sadrži više od C točaka iz S.*

$k = 1$

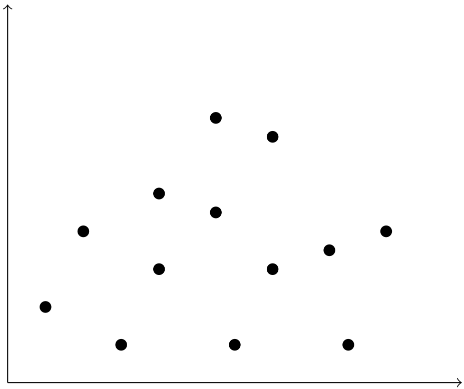
- N točaka na pravcu, zadani c i d
- dovoljno je za svaku točku provjeriti sljedećih c točaka



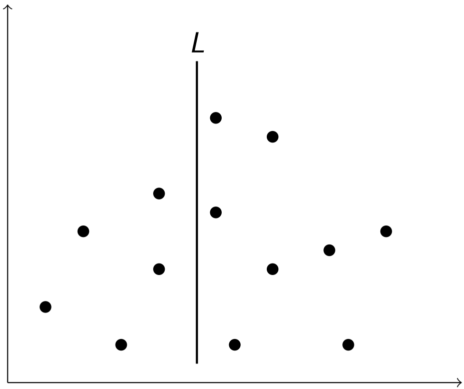
- sortiranje - $\mathcal{O}(N \lg N)$
- ispitivanje - $\mathcal{O}(N)$

Ukupno $\mathcal{O}(N \lg N)$

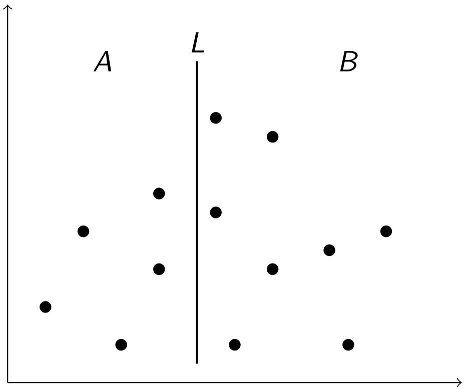
$k = 2$



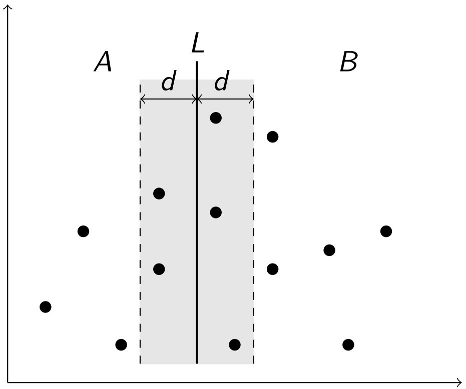
$k = 2$



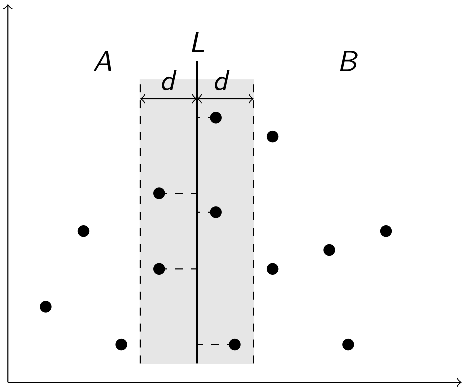
$k = 2$



$k = 2$



$k = 2$



$k = 2$

- Podijelimo skup S pravcem L “otprilike na pola” - particionirali smo S na podskupove A i B
- U skupovima A i B pronađemo parove bliskih točaka rekurzivnim pozivom algoritma

- Još treba pronaći parove kod kojih je jedna točka u A, a druga u B
- Svi takvi parovi nalaze se unutar pruge širine $2d$ oko L

$k = 2$

Projiciramo točke iz pruge na L

- projekcija čuva raspršenost
- udaljenost projiciranih točaka je manja ili jednaka originalnoj udaljenosti

Tražimo parove bliskih (projiciranih) točaka na pravcu

- složenost $\mathcal{O}(N \lg N)$
- ako smo sortirali točke na početku (npr. leksikografski), složenost je $\mathcal{O}(N)$
- treba još provjeriti da su točke prije projekcije bile bliske, te da je jedna točka iz A, a druga iz B

$k = 2$

Dobivamo rekurziju

$$T(N) = T(\lfloor N/2 \rfloor) + T(\lceil N/2 \rceil) + \mathcal{O}(N)$$

čije rješenje je

$$T(N) = \mathcal{O}(N \lg N).$$

$k = 3$

- Podijelimo skup S ravninom L na podskupove A i B otprilike jednake veličine
- Rekurzivno nađemo parove bliskih točaka u A i B
- Treba naći parove kod kojih je jedna točka iz A, a druga iz B - sve točke čija je udaljenost od L manja ili jednaka d projiciramo na L i rješavamo analogni problem u dimenziji $k = 2$
- Dobivamo rekurziju

$$T(N) = T(\lfloor N/2 \rfloor) + T(\lceil N/2 \rceil) + \mathcal{O}(N \lg N)$$

čije rješenje je

$$T(N) = \mathcal{O}(N \lg^2 N).$$

- Analogno postupamo i za dimenzije $k > 3$
- Matematičkom indukcijom dobivamo da je složenost algoritma za prostor dimenzije k jednaka

$$T(N) = \mathcal{O}(N \lg^{k-1} N)$$

Implementacija

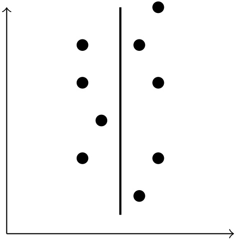
```
ulaz: Niz tocaka S sortiran leksikografski
izlaz: Parovi tocaka cija je udaljenost manja ili jednaka d

nadjiParove (S)
        ako u S postoje barem dvije tocke
                M = medijan tocaka u S obzirom na antileksikografski uredjaj

                za svaku tocku T iz S
                        ako je (T.y < M.y) ili (T.y = M.y i T.x < M.x)
                                stavi T u A
                        inace stavi T u B

                nadjiParove (A)
                nadjiParove (B)

                za svaku tocku T iz S
                        ako je |T.y - M.y| <= d
                                stavi T u L

                za svaku tocku T iz L
                        za svaku tocku P od sljedecih c tocaka
                                ako je dist(T, P) <= d i ako je
                                   (( T ∈ A  i  P ∈ B ) ili ( T ∈ B  i  P ∈ A ))
                                        ispisi (T, P)
```

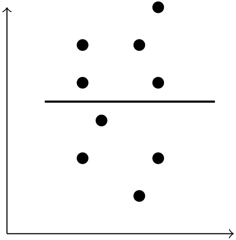
Implementacija

```
ulaz: Niz tocaka S sortiran leksikografski
izlaz: Parovi tocaka cija je udaljenost manja ili jednaka d

nadjiParove (S)
        ako u S postoje barem dvije tocke
                 M = medijan tocaka u S obzirom na antileksikografski uredjaj

                za svaku tocku T iz S
                        ako je (T.y < M.y) ili (T.y = M.y i T.x < M.x)
                                stavi T u A
                        inace stavi T u B

                nadjiParove (A)
                nadjiParove (B)

                za svaku tocku T iz S
                        ako je |T.y - M.y| <= d
                                stavi T u L

                za svaku tocku T iz L
                        za svaku tocku P od sljedecih c tocaka
                                ako je dist(T, P) <= d i ako je
                                   (( T ∈ A  i  P ∈ B ) ili ( T ∈ B  i  P ∈ A ))
                                           ispisi (T, P)
```

Može li brže?

- Dobrim odabirom pravca (hiperravnine) L možemo značajno smanjiti broj točaka koje treba projicirati
- Na taj način posljednji korak algoritma možemo provesti u vremenu manjem od linearnog





Može se pokazati da vrijedi sljedeće:

Teorem

Za svaki raspršen skup točaka S u k-dimenzionalnom prostoru postoji hiperravnina L sa sljedećim svojstvima:

(i) *moguće je pronaći L u linearnom vremenu*

(ii) *L dijeli skup S otprilike na pola*

(iii) *točaka udaljenih od L za $\leq d$ ima najviše $\mathcal{O}(N^{1-1/k})$.*

Korištenjem “dobre” hiperravnine, algoritam zadovoljava rekurziju

$$T_k(N) = T_k(\lfloor N/2 \rfloor) + T_k(\lceil N/2 \rceil) + \mathcal{O}(N),$$

čijim rješavanjem dobivamo vremensku složenost

$$T_k(N) = \mathcal{O}(N \lg N).$$

Sadržaj

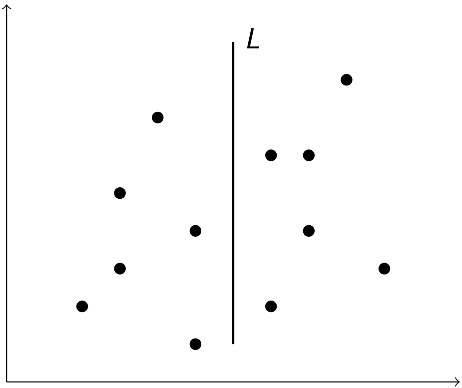
Najbliži par točaka

- Dan je skup S od N točaka u $\mathbb{R}^k$
- Treba naći najbliži par točaka iz skupa S

- $k = 1$ - sortiranje i sekvencijalno pretraživanje, složenost $\mathcal{O}(N \lg N)$
- $k \geq 2$ - višedimenzionalni "podijeli pa vladaj"

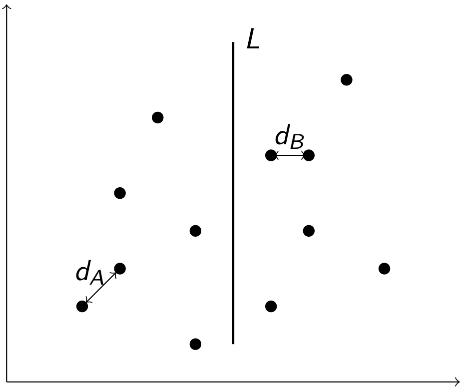
Najbliži par točaka, $k = 2$

- skup S podijelimo pravcem L na podskupove A i B
- rekurzivno nađemo najbliži par točaka u A i B; označimo odgovarajuće minimalne udaljenosti sa d_A i d_B



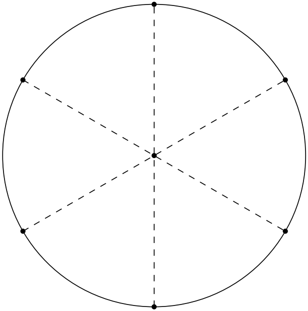
Najbliži par točaka, $k = 2$

- skup S podijelimo pravcem L na podskupove A i B
- rekurzivno nađemo najbliži par točaka u A i B; označimo odgovarajuće minimalne udaljenosti sa d_A i d_B



Najbliži par točaka, $k = 2$

- Ovo inducira raspršenost u A i B - svaka kugla polumjera d_A sadrži najviše 7 točaka iz A i svaka kugla polumjera d_B sadrži najviše 7 točaka iz B, pa svaka kugla polumjera $d = \min\{d_A, d_B\}$ sadrži najviše 14 točaka iz S



Najbliži par točaka, $k = 2$

- Još treba provjeriti postoji li par točaka čija je udaljenost manja od d takav da je jedna točka iz A, a druga iz B
- Točke iz pruge širine $2d$ oko L projiciramo na L i provodimo treći korak prethodnog algoritma, uz potrebne dodatne provjere
- Složenost (bez početnog sortiranja) je

$$T(N) = \mathcal{O}(N \lg^2 N)$$

Najbliži par točaka

U višim dimanzijama postupak je analogan:

- podijelimo S hiperravninom L na skupove A i B
- rekurzivno nađemo najbliži par točaka u A i B; minimalne udaljenosti su d_A i d_B redom
- sve točke udaljene od L za manje od $d = \min\{d_A, d_B\}$ projiciramo da L i tražimo parove točaka udaljenosti d - $\mathcal{O}(N \lg N)$

Najbliži par točaka

Dobivamo rekurziju

$$T_k(N) = T_k(\lfloor N/2 \rfloor) + T_k(\lceil N/2 \rceil) + \mathcal{O}(N \lg N),$$

čije rješenje je

$$T_k(N) = \mathcal{O}(N \lg^2 N).$$

- Algoritam možemo dodatno ubrzati odabirom "dobre" hiperravnine L - složenost $\mathcal{O}(N \lg N)$

Literatura

Jon Louis Bentley, *Multidimensional Divide-and-Conquer*, Communicationas of the ACM v.23, n.4 (April 1980), p.214-229.