

Algorithmik

Sebastian Iwanowski
FH Wedel

12. Vorlesungswoche

Algorithmik 12

Sweep-Verfahren

Transformation statisch d-dimensional \rightarrow dynamisch (d-1)-dimensional

d = 1: Line Sweep

1) Maximumsuche unter n Zahlen ($O(n)$)

2) Closest Pair:

Suche unter n Zahlen die beiden, die am dichtesten beieinander liegen. $O(n \log n)$

Preprocessing: Sortiere alle Zahlen $O(n \log n)$

Sweep: Scan von links nach rechts und merke das jeweils dichtesten Zahlenpaar $O(n)$

Referenzen zum Nacharbeiten und Vertiefen:

Klein, Kap. 2.2

Algorithmik 12

Sweep-Verfahren

Transformation statisch d-dimensional \rightarrow dynamisch (d-1)-dimensional

d = 2: Plane Sweep

3) Closest Pair:

Suche unter n Punkten die beiden, die am dichtesten beieinander liegen. $O(n \log n)$

Preprocessing: Sortiere alle Punkte nach x-Koordinate

Sweep: Scan von links nach rechts mit 2 vertikalen Geraden *left* und *right*.

Invarianten:


Horizontaler Abstand zwischen *left* und *right* ist gleich der minimalen Distanz des dichtesten Punktpaares links von *left*.

In *linecontent* sind alle Punkte zwischen *left* und *right* nach y-Koordinate geordnet enthalten.

Ereignisse und Aktionen:

left erreicht Punkt: Punkt wird gelöscht $O(\log n)$

right erreicht Punkt: Punkt wird in *linecontent* eingefügt und sein Abstand $O(\log n)$

nur konstant viele!  zu allen Punkten aus *linecontent* ausgerechnet, deren y-Koordinate um maximal der bisher gefundenen minimalen Distanz abweichen.

Referenzen zum Nacharbeiten und Vertiefen: Klein, Kap. 2.3.1

Algorithmik 12

Sweep-Verfahren

Transformation statisch d-dimensional \rightarrow dynamisch (d-1)-dimensional

Charakteristische Eigenschaften von Sweep-Verfahren:

- Scan über ausgewählte und sortierte x-Koordinaten (Ereignisse) von links nach rechts
- Sweep status structure (SSS) mit invarianten Eigenschaften
- Ereignisse werden statisch im Preprocessing sowie dynamisch bei der Aktualisierung des SSS berechnet.
- Schlafende Objekte: rechts von SSS, werden noch betrachtet
Aktive Objekte: in SSS, sind im Moment für Aktualisierungen relevant
Tote Objekte: links von SSS, müssen nie wieder betrachtet werden

Referenzen zum Nacharbeiten und Vertiefen:

Klein, Kap. 2.3.1

Algorithmik 12

Berechnung von Voronoi-Diagrammen mit Plane Sweep

Objekte der SSS:

- rechte vertikale Gerade L
- linke Wellenfront bestehend aus:
 - Parabelstücke von Bisektor $B(p,L)$, charakterisiert durch p und die anliegenden Parabelstücke nach oben und unten sowie die zugehörigen Spikes (s.u.)
 - Schnittpunkte zwischen angrenzenden Parabelstücken, geordnet nach y -Koordinate
 - Spikes: Bisektoren $B(p,q)$ für zwei benachbarte Referenzpunkte p und q
Jedes Wellenstück hat zwei benachbarte Spikes (außer erstem und letztem)

Lemma: Die Gesamtkomplexität der Wellenfront und damit auch der SSS ist $O(n)$

Referenzen zum Nacharbeiten und Vertiefen:

Klein, Kap. 6.3

Algorithmik 12

Berechnung von Voronoi-Diagrammen mit Plane Sweep

Ereignisse und Aktionen beim Sweep:

- Spike-Ereignis: Zugehöriges Wellenstück verschwindet.
- Punkt-Ereignis: Neues Wellenstück entsteht:
Neue Spikes und Spike-Ereignisse müssen berechnet werden.

Laufzeit: Ereignisaktualisierung in $O(\log n)$

$O(n)$ Ereignisse → **Gesamtlaufzeit:** $O(n \log n)$

Das ist optimal!

Referenzen zum Nacharbeiten und Vertiefen:

Klein, Kap. 6.3