

# Programmierpraktikum Diskrete Optimierung

## Thema: Effiziente Algorithmen für große Rechteckmengen

Homepage der Veranstaltung:

[http://www.or.uni-bonn.de/lectures/ss07/proprakt\\_ss07.html](http://www.or.uni-bonn.de/lectures/ss07/proprakt_ss07.html)

Ansprechpartner: Ulrich Brenner, E-Mail: [brenner@or.uni-bonn.de](mailto:brenner@or.uni-bonn.de)

### Allgemeine Bemerkungen:

In diesem Programmierpraktikum sollen effiziente Datenstrukturen und Algorithmen implementiert werden, mit denen sich große Mengen von Rechtecken verwalten lassen. Die Instanzen, auf denen die in diesem Praktikum erstellten Programme getestet werden sollen, stammen aus dem Entwurf von Bauplänen für höchstintegrierte Logikchips (VLSI-Chips). Diese Instanzen haben oft viele Millionen Elemente, daher sind für Algorithmen, mit denen diese Rechteckmengen bearbeitet werden sollen, nur lineare oder wenigstens fast lineare Laufzeiten (also bis auf einen polylogarithmischen Faktor) akzeptabel.

Die Rechtecke, die wir betrachten, sind achsenparallele Rechtecke im  $\mathbb{R}^2$ , d.h. ein Rechteck  $c$  ist gegeben als Punktmenge  $[x_{\min}(c), x_{\max}(c)] \times [y_{\min}(c), y_{\max}(c)]$ , wobei also  $(x_{\min}(c), y_{\min}(c))$  die Position der linken unteren Ecke und  $(x_{\max}(c), y_{\max}(c))$  die Position der rechten oberen Ecke ist.

**Die Programme in diesem Praktikum sind in ISO-C oder C++ zu schreiben.**

Es ist zunächst eine Einarbeitungsaufgabe zu bearbeiten. Anschließend wird es eine Zwischenbesprechung geben, in der die Ergebnisse der Einarbeitungsaufgabe und Fragen zu den weiteren Aufgaben besprochen werden sollen. Wenn Fragen oder Probleme auftauchen, können Sie sich natürlich auch sonst jederzeit melden.

### Aufgaben:

#### Einarbeitungsaufgabe:

In Abschnitt 8.4 von [2] wird ein *Sweepline-Verfahren* beschrieben, das für eine gegebene Menge von  $n$  Rechtecken in Zeit  $O(n \log n)$  den Inhalt der überdeckten Fläche bestimmt. Erstellen Sie eine möglichst effiziente Implementierung dieses Verfahrens. Natürlich sollen Sie dabei die angegebene asymptotische Laufzeit erreichen, aber darüber hinaus sollen Sie auch mit möglichst kleinen konstanten Faktoren auskommen. Insbesondere müssen Sie dazu eine Datenstruktur aufbauen, mit der Sie dynamisch eine Menge von Intervallen verwalten können. Eine nähere Beschreibung solcher Datenstrukturen findet sich auch in [1]. Die effiziente Verwaltung von Intervallen wird auch für die Bearbeitung der weiteren Teilaufgaben wichtig sein.

**Die Lösung der Einarbeitungsaufgabe muß spätestens bis zum 4. Mai 2007 abgegeben werden.** Spätestens bis zu diesem Zeitpunkt benötigen wir auch Ihre verbindliche Anmeldung. Es werden nur Lösungen von Teilnehmerinnen und Teilnehmern angenommen, die sich zuvor angemeldet haben.

### Weitere Teilaufgaben:

1. Implementieren Sie ein Sweepline-Verfahren, das zu zwei gegebenen Rechteckmengen  $M$  und  $K$  alle Paare  $(c, c') \in M \times K$  findet, für die die von  $c$  und  $c'$  überdeckten Flächen nichtleeren Schnitt haben. Ihr Verfahren soll Laufzeit  $O(n \log n + k)$  haben, wobei  $n = |M| + |K|$  und  $k$  die Zahl der gefundenen Paare ist. Ein solches Verfahren ist in Abschnitt 8.8 von [2] beschrieben. Erweitern Sie Ihr Programm so, daß es zu jedem gefundenen Paar den Schnitt der beiden Rechtecke bestimmt und in eine neue Rechteckliste  $L$  einfügt. Berechnen Sie auch den Inhalt der durch die Rechtecke in  $L$  überdeckten Fläche.
2. Implementieren Sie ein Sweepline-Verfahren für das folgende Problem. Zu einer gegebenen Menge  $M$  von Rechtecken wird eine Rechteckmenge  $K$  gesucht, so daß die Rechtecke in  $K$  dieselbe Fläche überdecken wie die in  $M$ , aber zusätzlich folgenden Bedingungen genügen:
  - Zu je zwei Rechtecken in  $K$  gibt es höchstens auf deren Rand Punkte, die von beiden überdeckt werden.
  - Es darf in  $K$  keine zwei Rechtecke  $c$  und  $c'$  geben, die gleiche  $y$ -Koordinaten haben (also  $[y_{\min}(c), y_{\max}(c)] = [y_{\min}(c'), y_{\max}(c')]$ ) und für die gilt  $x_{\min}(c) = x_{\max}(c')$ . Es darf also keine zwei Rechtecke mit gleicher Höhe geben, die direkt nebeneinander liegen. Ebenso darf es in  $K$  keine zwei Rechtecke  $c$  und  $c'$  geben, die gleiche  $x$ -Koordinaten haben (also  $[x_{\min}(c), x_{\max}(c)] = [x_{\min}(c'), x_{\max}(c')]$ ) und für die gilt  $y_{\min}(c) = y_{\max}(c')$ . Es darf also auch keine zwei Rechtecke mit gleicher Breite geben, die direkt untereinander liegen.

Außerdem darf  $K$  keine Rechtecke mit nichtleerem Inneren enthalten, wenn  $M$  keine solchen enthielt.

3. Schreiben Sie ein Programm, das im folgenden Sinn das Komplement der von zwei Rechteckmengen  $M$  und  $K$  überdeckten Flächen bestimmt: Wir betrachten die Menge aller Punkte  $X$ , die von Rechtecken in  $M$  aber nicht von solchen in  $K$  überdeckt werden. Gesucht ist dann der Abschluss von  $X$  (insbesondere sind dann also alle Ränder enthalten). Auch dieses Problem soll mit einem Sweepline-Verfahren unter Verwendung der Datenstrukturen zur Intervall-Verwaltung gelöst werden.
4. Die bisher geschriebenen Funktionen sollen nun benutzt werden, um eine Menge von Rechtecken so zu manipulieren, daß nur noch die großen überdeckten Flächen erhalten bleiben. Genauer gesagt: Zu einer gegebenen Menge  $M$  von Rechtecken wollen wir die Punkte bestimmen, deren euklidischer Abstand zum nächsten nicht überdeckten Punkt größer als ein gegebener Wert  $\delta > 0$  ist. Die Punktmenge soll wieder durch eine (möglichst kleine) Menge von Rechtecken, die genau diese Punkte überdecken, dargestellt werden.

Implementieren Sie das folgende Verfahren zur Lösung dieses Problems: Zunächst wählt man sich ein hinreichend großes zusätzliches Rechteck  $c$ , das insbesondere alle gegebenen Rechtecke enthält. Anschließend berechnet man im Sinne der vorigen Aufgabe alle Gebiete in  $c$ , die nicht von Rechtecken in  $M$  überdeckt werden (dargestellt als Menge  $K$  von Rechtecken). Die Rechtecke in  $K$  vergrößert man dann in alle vier Richtungen um  $\delta$  und berechnet wieder das Komplement dieser vergrößerten Rechtecke. Zuletzt muß man diese Rechtecke wieder um  $\delta$  vergrößern.

Auf den Testinstanzen, die Sie auf der Homepage der Veranstaltung herunterladen können, ist z.B. 100 ein vernünftiger Wert für  $\delta$  (aber natürlich sollen das Programm mit jedem positiven  $\delta$ -Wert funktionieren).

**Die Lösungen der Teilaufgaben sind spätestens bis zum 1. August 2007 abzugeben.**

## Datenformat und Testdaten:

Testdaten finden Sie (spätestens ab 19. Februar) auf der oben angegebenen Homepage des Praktikums. Die Rechtecklisten, die als Eingabe dienen, sind dabei als Textdateien gegeben. Die erste Zeile einer solchen Datei enthält die Zahl der Rechtecke, jede weitere Zeile kodiert genau ein Rechteck in der Form  $x_{\min}(c) \ x_{\max}(c) \ y_{\min}(c) \ y_{\max}(c)$ . Bei der Darstellung nicht-ganzzahliger Werte wird ein Dezimalpunkt verwendet, und die einzelnen Zahlen sind durch Leerzeichen getrennt. Die folgenden Zeilen kodieren z.B. eine zweielementige Rechteckliste, die aus den Rechtecken  $[-1.0, 1000.0] \times [5.0, 300.0]$  und  $[-1.0, 2500.0] \times [5.5, 2000.0]$  besteht:

```
2
-1.0 1000.0 5.0 300.0
-1.0 2500.0 5.5 2000.0
```

Die einzelnen Programme sollen so aufgerufen werden können, daß man ihnen den Namen von ein oder zwei Dateien mit Rechtecklisten übergibt (je nach Aufgabenstellung) und für die letzte Aufgabe den Wert von  $\delta$ . Wenn also `program` ein Programm zur Lösung der letzten Aufgabe ist, das die Datei `rectlist` einlesen soll und einen Wert von  $\delta = 5$  benutzen soll, sieht der Aufruf so aus:

```
program rectlist 5.0
```

## Literatur

- [1] S.W. Cheng, R. Janardan: *Efficient maintenance of the union of intervals on a line, with applications*. Journal of Algorithms, 12, 1991, 57–74.
- [2] F.P. Preparata, M.I. Shamos: *Computational Geometry*. Springer, 1988.