

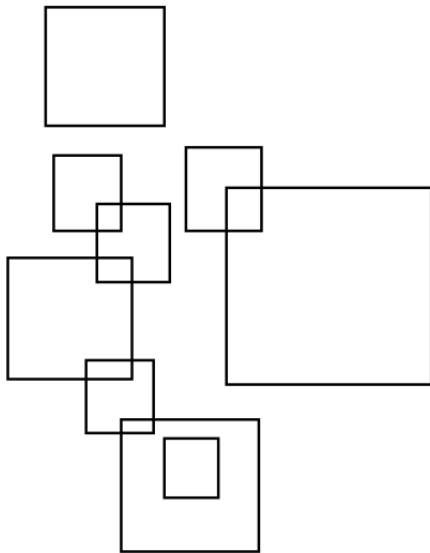
# Toepassingen van meetkunde in de informatica

## Practicum: Snijdende rechthoeken

In een vlak liggen  $N$  rechthoeken, met horizontale en verticale randen, waarvan telkens het linker-onderhoekpunt en het rechter-bovenhoekpunt gegeven zijn, d.w.z.  $lo_1, \dots, lo_N$  en  $rb_1, \dots, rb_N$  gegeven zijn. We willen alle snijpunten tussen de rechthoeken bepalen. Merk op: een rechthoek kan volledig binnen een ander rechthoek liggen: deze rechthoeken snijden elkaar niet.

Ontwerp een algoritme en schrijf vervolgens een programma dat dit probleem oplost:

1. een eenvoudig algoritme met rekencomplexiteit  $O(N^2)$ ,
2. een doorlooplijnalgoritme met een 'slechtste-geval' rekencomplexiteit  $O(N^2)$ ,
3. (*facultatief*) een doorlooplijnalgoritme met rekencomplexiteit  $O((N + S)\log_2(N))$ , met  $S$  het aantal snijpunten.



Tabel 1: een aantal snijdende vierkanten (dit zijn ook rechthoeken!)

## 1 Hoogniveau beschrijving van de algoritmen

Geef voor de twee of drie algoritmen een *hoog-niveau*<sup>1</sup> beschrijving.

Een doorlooplijnalgoritme maakt gebruik van een doorlooplijn die zich van links naar rechts door het vlak beweegt. Indien je bijhoudt welke rechthoeken “actief” zijn voor een bepaalde positie van de doorlooplijn (d.w.z. die de doorlooplijn snijden) kan je het aantal testen op snijding beperken. In het slechtste geval zal toch voor elke rechthoek een test op snijding met elke andere rechthoek nodig zijn.

Indien je gepaste gegevensstructuren gebruikt, kan de rekencomplexiteit beperkt worden tot  $O((N + S)\log_2(N))$ .

---

<sup>1</sup>d.w.z. op de manier waarop algoritmes in de cursus en de oefenzittingen worden voorgesteld, onafhankelijk van een specifieke programmeertaal

## 2 Implementatie

Je mag de implementatie maken in een programmeertaal naar keuze.

### 2.1 Algoritmen

Implementeer de twee of drie algoritmen om de snijpunten van de rechthoeken te bepalen. Voor de implementatie van de doorlooplijnstatus mag je gebruik maken van een voorgedefinieerde binaire zoekboom (bijvoorbeeld voor Java <http://algs4.cs.princeton.edu/32bst/BST.java.html>).

Je hoeft geen rekening te houden met *speciale gevallen*, maar je moet wel aangeven in uw verslag voor welk speciale gevallen (randgevallen) uw algoritme niet werkt.

### 2.2 Invoer en uitvoer

Maak een uitvoerbaar bestand waarmee de verschillende algoritmen eenvoudig getest kunnen worden. De invoer wordt gegeven als een txt-bestand dat op de volgende manier is opgebouwd:

- De eerste lijn bevat één getal: het nummer van het algoritme dat gebruikt moet worden 1, 2 of 3
- De volgende lijn bevat één getal: het aantal rechthoeken  $N$
- De volgende  $N$  lijnen zijn een opeenvolging van vier reële getallen  $lox_i$ ,  $loy_i$ ,  $rbx_i$ ,  $rby_i$  van elkaar gescheiden door een spatie. De getallen stellen de  $x$ - en  $y$ -coördinaten van respectievelijk het linker-onder-hoekpunt en het rechter-boven-hoekpunt van de  $i$ -de rechthoek voor.

```
1
5
0.200 0.200 0.300 0.600
0.500 0.400 0.700 0.600
0.400 0.600 0.600 0.800
0.350 0.600 0.500 0.900
0.200 0.700 0.900 0.800
```

Tabel 2: voorbeeld input.txt

De uitvoer van het gekozen algoritme moet weggeschreven worden naar een txt-bestand dat er als volgt moet uitzien (zie Tabel 2):

- Het bestand bevat slechts één regel: Dit algoritme is niet geïmplementeerd.  
of
- De eerste  $S$  lijnen, met  $S$  het aantal gevonden snijpunten bevatten telkens de  $x$ - en  $y$ -coördinaat van een snijpunt, gescheiden door een spatie.
- Een blanco lijn
- De laatste lijn bevat een getal dat de uitvoeringstijd weergeeft in milliseconden.

## 2.3 Experimenten

De geïmplementeerde algoritmen moeten gebruikt worden om enkele experimenten uit te voeren op willekeurige verzamelingen van rechthoeken.

1. Je werkt best met rechthoeken waarvan de  $x$ - en  $y$ -coördinaten van de hoekpunten tussen 0 en 1 liggen. Je kan hierbij gebruik maken van een randomgenerator die bewegende komma-getallen tussen 0 en 1 genereert. In Python levert de functie `random.random()` in de standaard module `random` zulke 'willekeurige' getallen. In Matlab kan je de functie `rand` gebruiken. In Java levert kan je de methode `random()` in de klasse `java.lang.Math` gebruiken.
2. Begin met een experiment met een beperkt aantal kleine rechthoeken, bv.  $N = 10$ , lengte van de zijden  $< 0.2$ . Verhoog dan het aantal rechthoeken zodat de rekencomplexiteit van de verschillende algoritmen duidelijk wordt. Ga na of de berekende oplossing correct is! Herhaal dan de experimenten met grotere rechthoeken, zodat er meer snijdingen optreden.

## 3 Verslag

Schrijf een verslag waarin je enkel de resultaten van je werk beschrijft (geen opgave herhalen, ...). Geef hierin zeker de volgende dingen weer:

1. Een hoogniveau beschrijving van al de uitgewerkte algoritmen
2. Een beschrijving van hoe je de experimenten hebt opgesteld en hoe je de correctheid van de code hebt nagegaan.
3. Een kritische en grondige *bespreking* van de resultaten van je experimenten. Verklaar ook de rekentijden.

## 4 Praktisch

De deadline van het project is **vrijdag 20 mei** om 14u.

### 4.1 Code

De code moet opgeladen worden op Toledo (door 1 persoon van het groepje indien je met twee gewerkt hebt). Zet al de code in één zip-bestand met jullie achternamen in de bestandsnaam, bv. `codeJanssensPeeters.zip`. Het zip-bestand moet zeker bevatten:

- het uitvoerbaar bestand (zie §2.2) waarmee wij je code kunnen testen. Het programma moet uitvoerbaar zijn op de computers van de computerklas A00.124 (<http://www.cs.kuleuven.be/cs/system/wegwijs/computerklas/>).
- een invoerbestand `invoerrechthoeken.txt` en het bijhorende uitvoerbestand `uitvoerrechthoeken.txt`
- een bestand `leesmij.txt` waarin kort beschreven staat hoe jouw programma moet uitgevoerd worden

Het verslag hoort niet thuis in dit zip-bestand !

## **4.2 Verslag**

Het verslag moet afgegeven worden in de studentenbrievenbus in 200A. Vergeet niet je naam, het vak en de bestemming (D. Roose) te vermelden! Indien je met twee hebt gewerkt, volstaat één verslag met beide namen op. Het verslag moet ook (apart) opgeladen worden op Toledo als pdf-bestand (door 1 persoon van het groepje indien je met twee gewerkt hebt).