

Toepassingen van meetkunde in de informatica

Robin Goots
Ward Schodts

2013 - 2014

Professor Dirk Roose

1 Hoogniveau beschrijving van de algoritmen

Algorithm 1 eenvoudig algoritme met rekencomplexiteit $O(N^2)$

```
Lijst L met alle cirkels
while Niet leeg L do
  Cirkel = L.NeemEnVerwijder
  for c in L do
    snijptn = Cirkel.BerekenSnijpunten(c)
    output.VoegToe(snijptn)
  end for
end while
return output
```

Voor het volgende algoritme stellen we elke cirkel voor als een lijnstuk met als links eindpunt het meest linkse punt van de cirkel en als rechts eindpunt het meest rechtse eindpunt van de cirkel.

Algorithm 2 doorlooplijnalgoritme met rekencomplexiteit $O(N^2)$

```
Lijst L: met alle punten, gesorteerd op het laagste x-coördinaat, daarna op het laagste
y-coördinaat en daarna komen linkse punten van een lijnstuk voor rechtse punten van een
lijnstuk.
Lijst C: met alle cirkels op een bepaald moment van een eventpoint, ongesorteerd.
for elk punt p in L do
  if p is het linkse eindpunt van een lijnstuk c then
    snijptn = Cirkel.BerekenSnijpunten(c)
    output.VoegToe(snijptn)
    VoegToe(C,c)
  end if
  if p is het rechtse eindpunt van een lijnstuk c then
    Verwijder(C,c)
  end if
end for
return output
```

2 Correctheid van de algoritmen verifiëren

Om de correctheid van de algoritmen te verifiëren hebben we verschillende stappen ondernomen. Deze stappen moeten in chronologische volgorde worden uitgevoerd omdat we assumpties uit voorgaande stappen gebruiken.

2.1 Enkele manuele testen

bepalen dat cirkels snijden

Onze eerste zorg was er zeker van zijn dat onze methode voor het bepalen dat cirkels snijden correct werkt. Met behulp van internet hebben we formules afgeleid om te bepalen of cirkels

Algorithm 3 complex doorlooptijnalgoritme met rekencomplexiteit $O((N + S)\log(N))$

Lijst L: met alle punten, gesorteerd op het laagste x-coördinaat daarna op het laagste y-coördinaat en daarna komen linkse punten van een segment voor rechtse punten.

Lijst S: met alle segmenten op een bepaald moment van een eventpoint, ongesorteerd.

```
for elk punt p in de gesorteerde lijst do
    if p is het linkse eindpunt van een cirkel c then
        snijptn = Cirkel.BerekenSnijpunten(c)
        output.VoegToe(snijptn)
        VoegToe(C,c)
    end if
    if p is het rechtse eindpunt van een cirkel c then
        Verwijder(C,c)
    end if
end for
return output
```

snijden. Na dat we deze formules afgeleid hebben, hebben we deze omgezet in Python code.

Bepalen van snijpunten van twee cirkels

Na dat we er zeker

Algoritme één

Hoewel de correctheid van algoritme één triviaal lijkt achten we het toch nodig om de correctheid hiervan te verifiëren. Dit hebben we gedaan door voor een klein aantal cirkels de snijpunten manueel uit te rekenen. Uit deze berekeningen bleek dat algoritme één correct werkt.