departement Computerwetenschappen



Projec

t 1: Ve

Toepassingen van meetkunde in de informatica

# Project : Bepaling van het Dichtste Puntenpaar

**Tibo Masselis r0638667**

**Barbara Ameloot r0669387**

Om te beginnen de hoogniveau beschrijving van de geïmplementeerde algoritmen.

Het eenvoudig algoritme:

rij

int d = infinity

While i = 0 < length(rij)

While j = i+1 < length(rij)

a = rij[i]

b = rij[j]

if Afstand(a,b) < d

dpp1 = a

dpp2 = b

d = Afstand(a,b)

j++

i++

return dpp1,dpp2,d

Doorlooplijnalgoritme, eerste variant:

rij

int d = infinity

While i = 0 < length(rij)

While (j = i+1 < length(rij)) AND (rij[j].x - rij[i].x < d)

a = rij[i]

b = rij[j]

if Afstand(a,b) < d

dpp1 = a

dpp2 = b

d = Afstand(a,b)

j++

i++

return dpp1,dpp2,d

Doorlooplijnalgoritme, tweede variant:

rij

t

int d = infinity

While (i = 0 < length(rij))

voegtoe(t, rij[i])

p = onder(t,rij[i])

While (p.y >= rij[i].y - d)

if (Afstand(p, rij[i]) < d)

dpp1 = p

dpp2 = rij[i]

d = Afstand(dpp1, dpp2)

for (p in t)

if (p.x < rij[i].x - d)

verwijder(t, p)

p = onder(t, p)

p = boven(t,rij[i])

While (p.y <= rij[i].y + d)

if (Afstand(p, rij[i]) < d)

dpp1 = p

dpp2 = rij[i]

d = Afstand(dpp1, dpp2)

for (p in t)

if (p.x < rij[i].x - d)

verwijder(t, p)

p = boven(t, p)

i++

return dpp1,dpp2,d

# 2. Vergelijking uitvoeringstijd eenvoudig algoritme en eerste variant doorlooplijnalgoritme

We willen de uitvoeringstijd van de eerste variant van het doorlooplijnalgoritme vergelijken met die van het eenvoudig algoritme voor het 2-dimensionale geval.

We hebben beide algoritmen getest op een oplopend aantal punten. Hieruit kregen we volgende resultaten:

Voor het eenvoudig algoritme:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aantal punten | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 | 8000 | 9000 | 10000 |
| Uitvoeringstijd  (in milliseconden) | 9 | 17 | 22 | 48 | 62 | 100 | 156 | 187 | 231 | 296 |

Voor de eerste variant van het doorlooplijnalgoritme:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aantal punten | 6250 | 12500 | 25000 | 50000 | 100000 | 200000 | 400000 | 800000 |
| Uitvoeringstijd  (in milliseconden) | 1 | 1 | 4 | 3 | 3 | 4 | 21 | 39 |

Hieruit blijkt duidelijk dat de eerste variant veel sneller is dan het eenvoudig algoritme voor het 2-dimensionale geval.