

Algoritmiek

Onderwerpen:

Sorteren (StraightSelectionSort, Bubblesort, InsertionSort)

Backtracking

Convex Hull

Recursie

Sorteren (QuickSort / MergeSort)

Sorteren (mbv een binaire boom)

'Greedy' algoritmes

Kortste pad (Dijkstra, Bellman-Ford, A* algoritme)

Alle kortste paden (Floyd-Warshall, Johnson)

Minimaal opspannende boom (Prim, Kruskal)

Kortste routes (handelsreizigesprobleem)

Maximale stroom (Ford Fulkerson)

Complexiteit

Heuristieken

Algoritmiek

Onderwerpen:

Sorteren (StraightSelectionSort, Bubblesort, InsertionSort)

Backtracking

Convex Hull

Recursie

Sorteren (QuickSort / MergeSort)

Sorteren (mbv een binaire boom)

'Greedy' algoritmes

Kortste pad (Dijkstra, Bellman-Ford, A* algoritme)

Alle kortste paden (Floyd-Warshall, Johnson)

Minimaal opspannende boom (Prim, Kruskal)

Kortste routes (handelsreizigesprobleem)

Maximale stroom (Ford Fulkerson)

Complexiteit

Heuristieken

Dijkstra's algoritme pseudocode



BerekenAfstand

...

Dijkstra's algoritme pseudocode

BerekenAfstand

```
Lees vertrek, bestemming, afstandentabel
van := vertrek
Zolang van <> bestemming
  naar := 1
  Zolang naar <= laatste
    Als (er is een pad tussen van en naar) en
      (afstand[van]+lengte[van,naar] < afstand[naar]) dan
        afstand[naar] := afstand[van]+lengte[van,naar]
    naar := naar+1
Zolang van <> bestemming
  naar := 1
  eerste := ∞
  Zolang naar <= laatste
    Als (niet bezocht[naar]) en (afstand[naar]<eerste)
      eerste := afstand[naar]
      van := naar
    naar := naar+1
```

Dijkstra's algoritme pseudocode

BerekenAfstand

Lees vertrek, bestemming, afstandentabel

van := vertrek

Zolang van <> bestemming

naar := 1

Zolang naar <= laatste

Als (er is een pad tussen van en naar) en

(afstand[van]+lengte[van,naar] < afstand[naar]) dan

afstand[naar] := afstand[van]+lengte[van,naar]

naar := naar+1

Zolang van <> bestemming

naar := 1

eerste := ∞

Zolang naar <= laatste

Als (niet bezocht[naar]) en (afstand[naar]<eerste)

eerste := afstand[naar]

van := naar

naar := naar+1

Dijkstra's algoritme pseudocode

BerekenAfstand

Lees vertrek, bestemming, afstandentabel

van := vertrek

Zolang van <> bestemming

naar := 1

Zolang naar <= laatste

Als (er is een pad tussen van en naar) en

(afstand[van]+lengte[van,naar] < afstand[naar]) dan

afstand[naar] := afstand[van]+lengte[van,naar]

naar := naar+1

Zolang van <> bestemming

naar := 1

eerste := ∞

Zolang naar <= laatste

Als (niet bezocht[naar]) en (afstand[naar]<eerste)

eerste := afstand[naar]

van := naar

naar := naar+1

Dijkstra's algoritme pseudocode

BerekenAfstand

```
Lees vertrek, bestemming, afstandentabel
van := vertrek
Zolang van <> bestemming
  naar := 1
  Zolang naar <= laatste
    Als (er is een pad tussen van en naar) en
      (afstand[van]+lengte[van,naar] < afstand[naar]) dan
        afstand[naar] := afstand[van]+lengte[van,naar]
    naar := naar+1
Zolang van <> bestemming
  naar := 1
  eerste := ∞
  Zolang naar <= laatste
    Als (niet bezocht[naar]) en (afstand[naar]<eerste)
      eerste := afstand[naar]
      van := naar
    naar := naar+1
```

Dijkstra's algoritme pseudocode

BerekenAfstand

```
Lees vertrek, bestemming, afstandentabel  
van := vertrek
```

```
Zolang van <> bestemming
```

```
    naar := 1  
    Zolang naar <= laatste  
        Als (er is een pad tussen van en naar) en  
            (afstand[van]+lengte[van,naar] < afstand[naar]) dan  
                afstand[naar] := afstand[van]+lengte[van,naar]  
        naar := naar+1
```

```
Zolang van <> bestemming
```

```
    naar := 1  
    eerste := ∞  
    Zolang naar <= laatste  
        Als (niet bezocht[naar]) en (afstand[naar]<eerste)  
            eerste := afstand[naar]  
            van := naar  
        naar := naar+1
```


Dijkstra's algoritme pseudocode

BerekenAfstand

Lees vertrek, bestemming, afstandentabel

van := vertrek

Zolang van <> bestemming

naar := 1

eerste := ∞

Zolang naar <= laatste

Als (er is een pad tussen van en naar) en

(afstand[van]+lengte[van,naar] < afstand[naar]) dan

afstand[naar] := afstand[van]+lengte[van,naar]

Als (niet bezocht[naar]) en (afstand[naar]<eerste)

eerste := afstand[naar]

volgende := naar

naar := naar+1

van := volgende

Dijkstra's algoritme pseudocode

ToonRoute

naar = bestemming

reis = naam[bestemming]

zolang naar <> vertrek

van = 1

zolang afstand[van] <> (afstand[naar] - lengte[van, naar])

van = van + 1

reis = naam[van] + reis

naar = van

Toon reis

Dijkstra's algoritme pseudocode

ToonRoute

```
naar = bestemming
```

```
reis = naam[bestemming]
```

```
zolang naar <> vertrek
```

```
    van = 1
```

```
    zolang afstand[van] <> (afstand[naar] - lengte[van, naar])
```

```
        van = van + 1
```

```
    reis = naam[van] + reis
```

```
    naar = van
```

Toon reis

Dijkstra's algoritme pseudocode

ToonRoute

naar = bestemming

reis = naam[bestemming]

zolang naar <> vertrek

van = 1

zolang afstand[van] <> (afstand[naar] - lengte[van, naar])

van = van + 1

reis = naam[van] + reis

naar = van

Toon reis

