

ALGORITHMS & DATA STRUCTURES

LES 6: GRAFEN

THEORIE

Actie	Hoofdstuk	2010	2012
Bestudeer	Graphs and paths	14	13

WEB

Filmpje waarin Dijkstra wordt uitgelegd:

<http://www.youtube.com/watch?v=8Ls1RqHCOPw&feature=channel>

THEORIE OPGAVEN

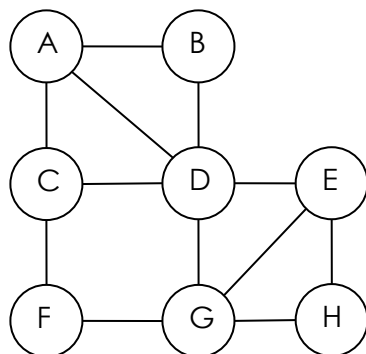
OPGAVE 1 (14.1)

Find the shortest weighted path from V_2 to all others in the graph shown in Figure 14.1 (2012 : 13.1)

OPGAVE 2 (14.3)

Find the shortest unweighted path from V_3 to all others in the graph shown in Figure 14.1 (2012 : 13.1)

OPGAVE 3



Voor bovenstaande graaf zijn de afstanden (beide richtingen hebben dezelfde waarde) tussen de vertices in onderstaande tabel te vinden:

Vertex 1	Vertex 2	Afstand	Vertex 1	Vertex 2	Afstand
A	B	4	C	F	3

C	D	3	B	D	2
D	E	1	D	G	3
F	G	2	E	H	4
G	H	2	A	D	1
A	C	1	E	G	1

Gevraagd: bereken volgens het algoritme van Dijkstra de kortste afstand van vertex C tot iedere andere vertex.

In welke volgorde worden de nodes gemarkeerd als "known" (of scratch=1)?

OPGAVE 4

Dezelfde opdracht als de vorige, maar nu begint het pad in Vertex F en geldt de volgende lijst met afstanden:

Vertex 1	Vertex 2	Afstand	Vertex 1	Vertex 2	Afstand
A	B	1	C	F	1
C	D	4	B	D	1
D	E	1	D	G	2
F	G	3	E	H	1
G	H	4	A	D	3
A	C	1	E	G	3

In welke volgorde worden de nodes gemarkeerd als "known" (of scratch=1)?

PRAKTIJK

OPDRACHT 1 (NODIG VOOR PRACTICUM)

- Implementeer de klasse `Vertex`. Deze klasse is een implementatie van `IVertex`.
- Voeg aan de klasse een `ToString()` method toe. De implementatie van deze method is als volgt:

- Voor een vertex V1 die niet verbonden is met andere vertices:

```
V1
```

- Voor een vertex V1 die verbonden is met V3 (cost=3) en V4 (cost=10):

```
V1 [ V3 (3) V4 (10) ]
```

- Voor dezelfde vertex, na het uitvoeren van een kortste pad algoritme waarbij een afstand van 5 tot de startvertex is berekend:

```
V1 (5) [ V3 (3) V4 (10) ]
```

- Alle whitespaces (spaties, tabs, newlines) kunnen worden genegeerd (dat doet de unittest ook).

- Implementeer de klasse `Graph` volgens de bijgeleverde interface.

d) Voeg aan deze klasse een method `ToString()` toe. Het resultaat van deze method is de uitvoer van de `ToString()` methods van de vertices.

- Tip: omdat whitespace wordt genegeerd, zou je voor de leesbaarheid de vertices ook kunnen scheiden door een newline. De uitvoer voor de graaf uit figuur 14.1 (1012: figuur 13.1) uit het boek zou dus zijn:

```
V0 [ V1(2) V3(1) ]
V1 [ V3(3) V4(10) ]
V2 [ V0(4) V5(5) ]
V3 [ V2(2) V5(8) V6(4) V4(2) ]
V4 [ V6(6) ]
V5
V6 [ V5(1) ]
```

OPDRACHT 2

Implementeer met behulp van de code uit het boek het ongewogen kortste pad:

```
public void Unweighted(string name)
```

Voor je algoritme kun je gebruik maken van de ingebouwde klasse `Queue` (in `System.Collections.Generic`). Voer het algoritme uit op de graaf van figuur 14.1 (2012: figuur 13.1) uit het boek. Voor de implementatie van `Unweighted()` kun je eventuele waarden van `cost` negeren.

Schrijf zelf unittests om te kijken of je algoritme werkt. Je zou kunnen kijken of de `ToString()` van de graaf na de aanroep van `Unweighted()` het verwachte resultaat geeft:

```
V0(0) [ V1(2) V3(1) ]
V1(1) [ V3(3) V4(10) ]
V2(2) [ V0(4) V5(5) ]
V3(1) [ V2(2) V5(8) V6(4) V4(2) ]
V4(2) [ V6(6) ]
V5(2)
V6(2) [ V5(1) ]
```

Breid `DSBuilder.cs` uit met andere bomen en schrijf ook andere unittests (wat gebeurt er bijvoorbeeld als niet alle nodes bereikt kunnen worden?).

OPDRACHT 3

Implementeer met behulp van de code uit het boek het gewogen kortste pad:

```
public void Dijkstra(string name)
```

C# biedt niet een ingebouwde klasse voor een `PriorityQueue`. Hier moet je zelf iets voor verzinnen. Je zou de vertices in een `List` of `LinkedList` kunnen bewaren en steeds de kleinste hieruit selecteren ($O(N)$). Een echte `priorityqueue` maken is natuurlijk nog mooier.

Voeg unittests toe voor het testen van je algoritme (zie ook hierboven).

OPDRACHT 4

Breid de graaf-klasse uit met de method `IsConnected()`. Deze method geeft terug of een gerichte graaf wel of niet verbonden (zie voor definitie <http://mathworld.wolfram.com/ConnectedGraph.html>) is.

Breid de unittests uit om je method te testen.