

2

Floyd

Inleiding

Download het bestand BG_Les9.zip van Toledo. Importeer dit bestand in Eclipse.

In deze oefenzitting leer je het algoritme van Floyd te implementeren in Java.

De eerste oefening is op papier. Hiermee leer je het algoritme goed begrijpen. Verder kan je deze papieren versie gebruiken om je zelfgeschreven code te testen. Houd de oplossing dus goed bij!

Oefening 2.1

Maak deze oefening op papier.

Een graaf bestaat uit 6 knooppunten en wordt gegeven door zijn gewichtenmatrix

$$D^{(0)} = \begin{bmatrix} 0 & 7 & 3 & \infty & 5 & \infty \\ 2 & 0 & \infty & 12 & \infty & \infty \\ \infty & 3 & 0 & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 & \infty & 6 \\ 5 & \infty & \infty & \infty & 0 & 2 \\ \infty & \infty & \infty & 1 & \infty & 0 \end{bmatrix}$$

1. Teken het netwerk.
2. Gebruik de methode van Floyd om de kortste paden tussen de verschillende punten van het netwerk te berekenen.

Oefening 2.2

Bestudeer de klasse `WeightedGraph` die je in Eclipse importeerde.

1. In tegenstelling tot de klasse `Graph` die we schreven bij BFS, is de instantieveranderlijke nu een matrix van `double` en niet van `boolean`. Waarom?

2 Floyd

2. Wanneer is een gewichtenmatrix geldig?

Oefening 2.3

Schrijf de methode `int[][] findDistances()`. Deze methode berekent de pointermatrix P .

- De (lege) pointermatrix wordt gedeclareerd.
- Er wordt een kloon van de gewichtenmatrix gemaakt. Dit is een kopie die verwijst naar een andere geheugenplaats. Deze kloon is de D -matrix van de cursustekst.
- Voeg de knopen één voor één toe als tussenstation en pas de D -matrix aan zoals aangegeven in formule (2.3). Update de pointermatrix.

Verwachte uitvoer:

```
p_matrix:
0 0 4 0 4
5 0 0 0 4
5 5 0 0 4
5 5 0 0 0
0 1 4 1 0
```

Oefening 2.4

Schrijf de methode `ArrayList<Integer> getShortestPath(int i, int j, int[][] path)`. Deze methode berekent uit de pointermatrix $path$ het kortste pad tussen knoop i en knoop j .

In deze oefening geldt dezelfde opmerking over de nummering van de knopen als we maakten in oefening 1.4. Het knooppunt dat laatst toegevoegd werd als tussenstation bij de berekening van het kortste pad tussen knooppunten i en j , is $path[i - 1][j - 1]$.

(Deel van) de verwachte uitvoer:

Kortste paden:

Er is geen pad van 1 naar 1

Kortste pad van 1 naar 2 lengte = 0 via : [1, 2]

Kortste pad van 1 naar 3 lengte = 0 via : [1, 4, 3]

Kortste pad van 1 naar 4 lengte = 0 via : [1, 4]

Kortste pad van 1 naar 5 lengte = 0 via : [1, 4, 5]

Kortste pad van 2 naar 1 lengte = 0 via : [2, 4, 5, 1]

Er is geen pad van 2 naar 2

Kortste pad van 2 naar 3 lengte = 0 via : [2, 3]

Kortste pad van 2 naar 4 lengte = 0 via : [2, 4]
Kortste pad van 2 naar 5 lengte = 0 via : [2, 4, 5]

Kortste pad van 3 naar 1 lengte = 0 via : [3, 4, 5, 1]
Kortste pad van 3 naar 2 lengte = 0 via : [3, 4, 5, 1, 2]
Er is geen pad van 3 naar 3
Kortste pad van 3 naar 4 lengte = 0 via : [3, 4]
Kortste pad van 3 naar 5 lengte = 0 via : [3, 4, 5]

Oefening 2.5

Schrijf de methode `int berekenLengte(ArrayList<Integer> pad)`. Invoer is pad: een op-eenvolging van knopen die resulteert in het kortste pad tussen eerste en laatste element van pad. Uitvoer is de lengte van dit pad. Deze wordt berekend met behulp van de instantieverschillende gewichtenmatrix. (Deel van) de verwachte uitvoer:

Kortste paden:
Er is geen pad van 1 naar 1
Kortste pad van 1 naar 2 lengte = 1 via : [1, 2]
Kortste pad van 1 naar 3 lengte = 3 via : [1, 4, 3]
Kortste pad van 1 naar 4 lengte = 1 via : [1, 4]
Kortste pad van 1 naar 5 lengte = 4 via : [1, 4, 5]

Kortste pad van 2 naar 1 lengte = 8 via : [2, 4, 5, 1]
Er is geen pad van 2 naar 2
Kortste pad van 2 naar 3 lengte = 3 via : [2, 3]
Kortste pad van 2 naar 4 lengte = 2 via : [2, 4]
Kortste pad van 2 naar 5 lengte = 5 via : [2, 4, 5]

Kortste pad van 3 naar 1 lengte = 10 via : [3, 4, 5, 1]
Kortste pad van 3 naar 2 lengte = 11 via : [3, 4, 5, 1, 2]
Er is geen pad van 3 naar 3
Kortste pad van 3 naar 4 lengte = 4 via : [3, 4]
Kortste pad van 3 naar 5 lengte = 7 via : [3, 4, 5]

Kortste pad van 4 naar 1 lengte = 6 via : [4, 5, 1]
Kortste pad van 4 naar 2 lengte = 7 via : [4, 5, 1, 2]
Kortste pad van 4 naar 3 lengte = 2 via : [4, 3]
Er is geen pad van 4 naar 4
Kortste pad van 4 naar 5 lengte = 3 via : [4, 5]

Kortste pad van 5 naar 1 lengte = 3 via : [5, 1]
Kortste pad van 5 naar 2 lengte = 4 via : [5, 1, 2]

2 Floyd

Kortste pad van 5 naar 3 lengte = 6 via : [5, 1, 4, 3]

Kortste pad van 5 naar 4 lengte = 4 via : [5, 1, 4]

Er is geen pad van 5 naar 5