

Kortste pad met dynamisch programmeren

Op blz. 89–91 van het boek wordt een algoritme beschreven voor het berekenen van het kortste pad in een gerichte acyclische graaf. Dit algoritme maakt gebruik van dynamisch programmeren.

a Geef een beschrijving in pseudo-code van dit algoritme. Je mag hierbij aannemen

- dat de graaf $N \geq 1$ knopen bevat
- dat de buren van een knoop X eenvoudig af te lopen zijn met behulp van de volgende constructie:

```
for alle buren  $Y$  van  $X$  do
{
}
```

- dat het gewicht van de tak van knoop X naar knoop Y te verkrijgen valt met behulp van de functie $\text{Gewicht}(X,Y)$
- dat de knopen in de graaf al geordend zijn in een array Sort (op posities $1 \dots N$), zó dat er alleen takken van links naar rechts lopen. Meer precies: als er een tak is van knoop X naar knoop Y , dan is $X = \text{Sort}[i]$ en $Y = \text{Sort}[j]$, met $1 \leq i < j \leq N$.

Het is de bedoeling dat het algoritme de afstand van de eerste knoop (in $\text{Sort}[1]$) tot de laatste knoop (in $\text{Sort}[N]$) berekent. Het doet dit door voor alle knopen (in omgekeerde volgorde, van N tot en met 1 in het array Sort) de afstand tot $\text{Sort}[N]$ te berekenen.

b Pas de pseudo-code uit het vorige onderdeel aan, zodat niet alleen de afstand van het kortste pad, maar ook het kortste pad zelf wordt bepaald.