Opdracht 3, Algoritmiek

Jenny Vermeltfoort, s3787494, groep PO3_29

15 mei 2025

Uitleg

Dit verslag omschrijft een implementatie die gebruikt maakt van dynamisch programmeren met memoization. De probleemstelling is het vinden van de meest goedkoopste route naar een eindbestemming. Een auto heeft een bepaalde tankcapaciteit en langs de route bestaan er n stations. Ieder station heeft een eigen prijs en een limiet op de hoeveelheid liter wat getankt kan worden.

Al de gegevens voor de puzzel worden uitgelezen uit een tekst bestand, met de volgende layout:

```
<capaciteit auto>
<aantal stations>
<x coordinaat station 0> <tankcapaciteit station 0> <literprijs station 0>
<x coordinaat station 1> <tankcapaciteit station 1> <literprijs station 1>
...
<x coordinaat station n> <tankcapaciteit station n> <literprijs station n>
<x coordinaat eindbestemming>
```

Implementatie

De functie $f : (i, j) \implies l_g$ associeert een station i en een tankinhoud j aan de goedkoopste kosten l_g. De hoeveelheid liter is; of volledig getankt bij station i; of gedeeltelijk meegenomen vanuit station $i-1$ en getankt bij station i, of volledig meegenomen vanuit station $i-1$.

Dit kan iteratief berekend worden door deze stelling op te delen in vier stukken. Zo omschrijft de functie 1 een deel waarbij station i goedkoper - $p_i < p_{i-1}$ - is als station $i-1$. Hierbij de volgende parameters:

i. sc_i de tank capaciteit van station i,

ii. $c_i = \min(c_{i-1} - a_i + sc_i, ac)$ de maximale capaciteit die de auto kan bereiken bij station i,

iii. a_i de afstand van station $i-1$ tot station i,

iv. ac de maximale tankcapaciteit van de auto,

v. $m_1 = a_i + j - sc_i$, $m_2 = a_i + j$ de hoeveelheid liter die wordt meegenomen vanuit het vorige station,

vi. p_i de liter prijs bij station i.

Functie 1 omschrijft het onderdeel waar het station i duurder is - $p_1 \geq p_{i-1}$. Neem aan de base case: $f(0, 1 \ldots j) = \infty$, $f(i, ac \ldots \infty) = \infty$ en $f(0,0) = 0$ en de afstand van station $i = 1$ is 0. Bij de functies neem $j \in \{0, 1, 2, \ldots c_i\}$, $i \in \{1, 2, 3, \ldots n\}$, en n het aantal stations.

$$
\begin{aligned}
p_i < p_{i-1} &\implies \begin{cases} f(i,j) = f(i-1, a_i) + j * p_i & \text{wanneer } 0 \ldots sc_i \\ f(i,j) = f(i-1, a_i + j - sc_i) + sc_i * p_i & \text{wanneer } (sc_i + 1) \ldots c_i \end{cases} \\
p_i \geq p_{i-1} &\implies \begin{cases} f(i,j) = f(i-1, a_i + j) & \text{wanneer } 0 \ldots (c_{i-1} - a_i) \\ f(i,j) = f(i-1, c_{i-1}) + (j - c_{i-1} - a_i) * p_i & \text{wanneer } (c_{i-1} - a_i + 1) \ldots c_i \end{cases}
\end{aligned} \tag{1}
$$

We nemen $l_g = sc_i$ en $l_d = c_{i-1} - a_i$. De volgende partiële functies worden geïmpliceerd uit de voorstaande formules:

$$\begin{aligned}
f_1 =& f(i-1, a_i) + j * p_i & j \leq l_g \vee l_d & \quad \textit{tank bij station } i. \\
f_2 =& f(i-1, c_{i-1}) + (j - l_d) * p_i & l_d < j \leq l_g & \quad \textit{mee genomen station } i-1, \textit{ tank bij station } i. \\
f_3 =& f(i-1, a_i + j) & l_g < j \leq l_d & \quad \textit{neem mee van station } i-1. \\
f_4 =& f(i-1, 1 + j - l_g) + l_g * p & l_g \vee l_d < j & \quad \textit{getankt bij station } i, \textit{ neem mee van station } i-1.
\end{aligned} \tag{2}$$

Jokers

We updaten de functie f naar $f_k : (i, j, k) \implies KJ_g$ deze associeert een station i, een tankinhoud j, en een k hoeveelheid gebruikte jokers aan de goedkoopste kosten KJ_g. De goedkoopste prijs word gevonden door bij het station $i-1$ een joker te gebruiken en de prijs van station i te gebruiken, of door bij station $i-1$ geen joker te gebruiken en bij station i een joker te gebruiken - dit in het geval dat er één joker in bezit is. Wanneer er meer als één joker in bezit is, dat wordt de hoeveelheid te gebruiken jokers gelimiteerd door de station index i, dus $k \in \{0, 1, 2, \ldots (i \vee n_k)\}$, met n_k het aantal jokers in bezit. Algemeen wordt de goedkoopste kosten f_k bij tankinhoud j bij een hoeveelheid jokers k, met j_t de hoeveelheid liter getankt bij station i, en j_m de hoeveelheid liter meegenomen vanuit het vorige station, uitgedrukt in:

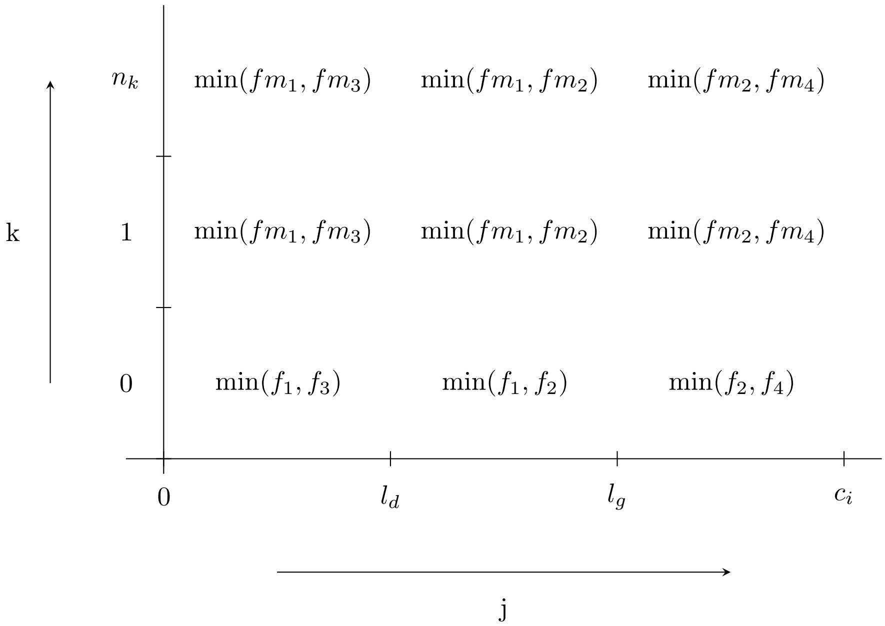
$$f_k(i, j, k) = \min(f_k(i-1, j_m, k-1) + \mathrm{ceil}(j_t * p_i * 0.5), f_k(i-1, j_m, k) + j_t * p_i) \tag{3}$$

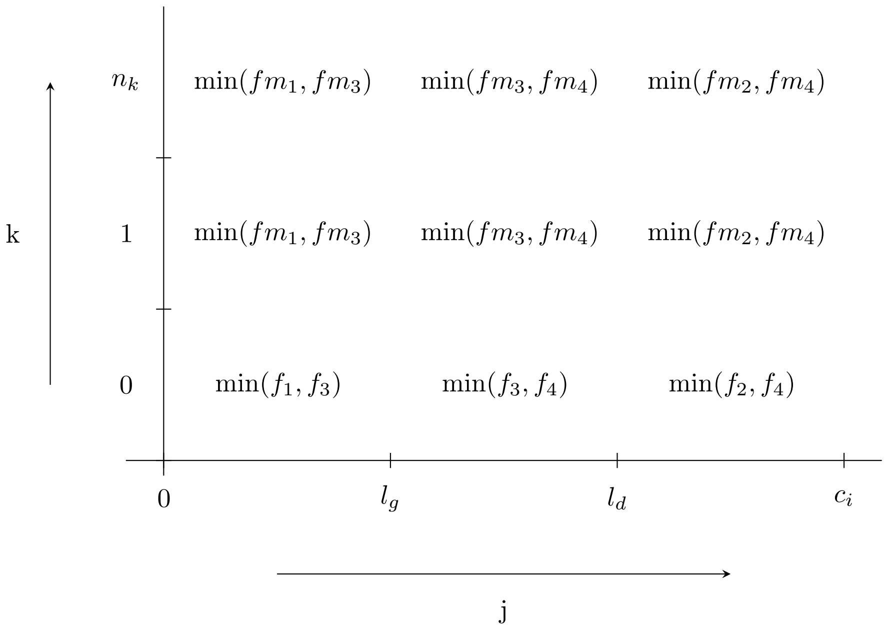
De volgende partiële functies worden geïmpliceerd uit de voorstaande formules:

$$\begin{aligned}
f_{k1} =& f(i-1, a_i, k) + j * p_i \\
f_{k2} =& f(i-1, c_{i-1}, k) + (j - l_d) * p_i \\
f_{k3} =& f(i-1, a_i + j, k) \\
f_{k4} =& f(i-1, 1 + j - l_g, k) + l_g * p \\
g_{k1} =& f(i-1, a_i, k-1) + \mathrm{ceil}(j * p_i * 0.5) \\
g_{k2} =& f(i-1, c_{i-1}, k-1) + \mathrm{ceil}((j - l_d) * p_i * 0.5) \\
g_{k3} =& f(i-1, a_i + j, k-1) \\
g_{k4} =& f(i-1, 1 + j - l_g, k-1) + \mathrm{ceil}(l_g * p * 0.5)
\end{aligned} \tag{4}$$

De limieten l_g en l_d waarmee j wordt begrenst zijn het makkelijkst uit de drukken met de volgende grafieken, hierbij de minima:

$$
\begin{aligned}
fm_1 =& \min(f_{k1}, g_{k1}) \\
fm_2 =& \min(f_{k2}, g_{k2}) \\
fm_3 =& \min(f_{k3}, g_{k3}) \\
fm_4 =& \min(f_{k4}, g_{k4})
\end{aligned}
\tag{5}
$$





We nemen aan dat $f_k(i, j, -1) = \infty$, waarmee dit alles leidt tot de volgende definitie van f_k:

$$
f_k(i,j,k) = \begin{cases}
\min(fm_1, fm_3) & j \leq l_g \wedge l_d \\
\min(fm_3, fm_4) & l_g < j \vee l_d \\
\min(fm_1, fm_2) & l_d < j \vee l_g \\
\max(fm_2, fm_4) & l_d \wedge l_g < j
\end{cases}
\tag{6}
$$

Oplossing

Om te weten wat de strategie is om de optimale oplossing te behalen moeten we weten hoeveel liter er op station i is getankt en of er een joker is gebruikt. We weten dat een joker is gebruikt wanneer $fk(i,j,k) = g_{kn}$; hier g_{kn} een van de functies in 4. Hieronder wordt beschreven hoeveel liter er is getankt:

$$\begin{array}{rl} j & fm_1 < fm_3 \wedge j \leq l_g \wedge l_d \\ 0 & fm_3 < fm_1 \wedge j \leq l_g \wedge l_d \\ 0 & fm_3 < fm_4 \wedge l_g < j \vee l_d \\ l_g & fm_4 < fm_3 \wedge l_g < j \vee l_d \\ j & fm_1 < fm_2 \wedge l_d < j \vee l_g \\ j - l_d & fm_2 < fm_1 \wedge l_d < j \vee l_g \\ j - l_d & fm_2 < fm_4 \wedge l_d \wedge l_g < j \\ l_g & fm_4 < fm_2 \wedge l_d \wedge l_g < j \end{array} \tag{7}$$

Of een joker is gebruikt wordt bijgehouden in een functie $h_k : (i,j,k) \implies J$ welke een station i, inhoud j en joker k associeert met of er een joker is gebruikt J. De hoeveelheid liter getankt worden bijgehouden in een functie $h_j : (i,j,k) \implies L$ welke een station i, inhoud j en joker k associeert met een hoeveelheid getankt L.

De optimale strategie wordt gevonden door van beneden naar boven over het grid te lopen, met $o_j(i)$ de strategisch beste hoeveelheid te tanken en $o_k(i)$ strategisch een joker te gebruiken. Hierbij als base case $h_{j/k}(as, a_{as}, k_n)$, vervolgens over $i \in \{(as-1)\ldots 1\}$ gelopen.

$$o_{j/k}(i) = h_{j/k}(i, j - h_j(i+1,j,k) + a_i, k - 1 * h_k(i+1,j,k)) \tag{8}$$

Tijdcomplexiteit

De worst-case tijdcomplexiteit van het bottum-up algoritme is $O(n * c * k)$. Deze tijdcomplexiteit wordt bepaald door de hoeveelheid $n * c * k$ cellen. In het worst-case scenario wordt elke cel gevuld met een waarde. Elke cel wordt echter maar één keer gevuld - de tak van de iteratie boom wordt afgekapt wanneer een cel al is gevuld. In worst-case scenario wordt dus elke cel maximaal één keer bezocht.

Resultaten

Zie appendix A voor de inhoud van de tesktbestand.

Voorbeeld 1

Tankcapaciteit van 100 liter, 20 stations, reis vanaf $x = -55$ tot $x = 373$ met max prijs 120 en max capaciteit station 90.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
j	18	19	51	27	0	0	74	0	0	25	60	54	56	30	14	0	0	0	0	0
k							✓				✓									

Tabel 1: Strategie voorbeeld 1 met twee jokers, totale kosten: 11481.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
j	28	19	51	27	6	12	46	0	0	25	60	54	56	30	14	0	0	0	0	0
k	✓		✓		✓	✓	✓			✓	✓	✓	✓		✓					

Tabel 2: Optimale strategie met tien jokers, totale kosten: 7870.

Voorbeeld 2

Tankcapaciteit van 100 liter, 13 stations, reis vanaf $x = 23$ tot $x = 253$ met max prijs 69 en max capaciteit station 39.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
j	25	9	30	35	2	8	37	32	29	10	0	13	0
k							✓						

Tabel 3: Voorbeel 2 met één joker, minimale kosten: 4424.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
j	25	9	30	35	2	8	37	32	29	10	0	13	0
k	✓	✓	✓		✓	✓	✓	✓	✓	✓			

Tabel 4: Strategie voorbeeld 2 met negen jokers gebruikt, minimale kosten: 2852.

Voorbeeld 3

Tankcapaciteit van 10 liter, 6 stations, reis vanaf $x = -30$ tot $x = 0$ met max prijs 50 en max capaciteit station 20.

n	0	1	2	3	4	5
j	7	0	0	10	4	9

Tabel 5: Strategie voorbeeld 3 zonder jokers, minimale kosten: 809.

n	0	1	2	3	4	5
j	7	0	0	10	4	9
k	✓	✓	✓	✓	✓	✓

Tabel 6: Strategie voorbeeld 3 met zes jokers, minimale kosten: 405.

Appendices

Voorbeelden

Voorbeeld 1

```
100
20
-55 28 50
-38 19 2
-22 51 22
-18 27 9
10 6 86
21 12 44
39 74 97
75 86 115
97 77 99
103 25 33
119 60 40
192 54 8
264 56 33
272 30 12
304 88 66
317 77 77
325 47 118
346 18 119
359 71 96
368 59 76
373
```

Voorbeeld 2

```
100
13
23 25 18
28 9 32
69 30 13
80 35 3
83 2 55
90 8 61
96 37 60
163 32 17
190 29 16
192 23 41
214 12 59
240 35 5
245 36 39
253
```

Voorbeeld 3

```
10
6
-30 9 23
-28 14 33
-26 13 45
-23 16 15
-15 4 21
-9 15 46
0
```