Het twee kannen probleem

Hoofdstuk 1

Inleiding

In dit artikel bespreken we problemen waarin men beschikt over 2 lege kannen, zonder maatstreepjes. Verder is er een kraan waarmee men de kannen kan vullen en een gootsteen waarin men de kannen kan leeggieten. We aanvaarden volgende handelingen :

1. Een kan volledig leeggieten.
2. Een kan helemaal vullen met de kraan.
3. Water van de ene kan overhevelen in de andere kan totdat de ene helemaal leeg is of de andere helemaal vol.

Het artikel bestaat uit twee delen. In een eerste deel gaan we op verkenning en geven we gepaste notaties aan de begrippen. in het tweede deel proberen we een algemeen resultaat neer te zetten en werken we nog een paar voorbeelden uit. We geven ook een mogelijke oplossing in Python.

Hoofdstuk 2

Het twee kannen probleem

2.1 Op verkenning met 3 liter en 5 liter

- Stel dat je beschikt over 2 lege kannen, één van 5 liter en één van 3 liter. Kan je hiermee juist 4 liter afpassen?
  - Vul de kan van 3 liter.
  - Giet ze over in de kan van 5 liter.
  - Vul opnieuw de kan van 3 liter.
  - Giet over tot de kan van 5 liter vol is. er blijft dan nog 1 liter over in de kleine kan.
  - Giet de grote kan leeg.
  - Giet die ene liter uit de kleine kan over in de grote kan.

  - Vul de kan van 3 liter
  - Giet die kan over in de grote kan. In die kan zit nu de gevraagde 4 liter.

- We gaan eerst op zoek naar een betere schrijfwijze van zo een proces. Noteren we de inhoud van de kannen in een koppel, met eerst de inhoud van de 3 liter kan en dan die van 5 liter. Het vorige proces wordt dan genoteerd door volgend rijtje, waarbij het cijfer boven de pijltjes geeft aan welke regel we gebruiken.
  $(0,0) \xrightarrow{2} (3,0) \xrightarrow{3} (0,3) \xrightarrow{2} (3,3) \xrightarrow{3} (1,5) \xrightarrow{1} (1,0) \xrightarrow{3} (0,1) \xrightarrow{2} (3,1) \xrightarrow{3} (0,4)$ Merk op dat we stap 2 (vullen van de 3 liter kan) 3 keer uitvoeren, stap 1 (legen 5 liter kan) slechts 1 keer e, spat 3 (overhevelen) 4 keer. Een merkwaardigheid: $3 * 3 - 1 * 5 = 4$.

- We kunnen ons hierbij een paar vragen stellen:

  1. Is dit de enige manier?
  2. De snelste manier?
  3. Kan je alle inhouden tussen 1 en 5 zo verkrijgen?
  4. Wat met andere begin inhouden?

- Er is zeker nog een nadere mogelijkheid:? Ze telt 1 stap minder dan de vorige methode.
  $(0,0) \xrightarrow{2} (0,5) \xrightarrow{3} (3,2) \xrightarrow{1} (0,2) \xrightarrow{3} (2,0) \xrightarrow{2} (2,5) \xrightarrow{3} (3,4) \xrightarrow{1} (0,4)$

  Stap 2 en stap 1 worden 2 keer uitgevoerd, stap 3 drie keer. Ook hier geldt: $2 * 5 - 2 * 3 = 4$. Een toeval?

- Bij het bepalen van zo een stappenplan moeten we wel rekening houden met de efficiëntie van de oplossing. Het is inefficiënt om:

  - Een handeling die je gedaan hebt, vervolgens omgekeerd te doen.
  - Een deelsgevulde kan leeg gieten.
  - Een deelgevulde kan verder vullen aan de kraan.
  - Als je het del bereikt hebt nog verder gaan.

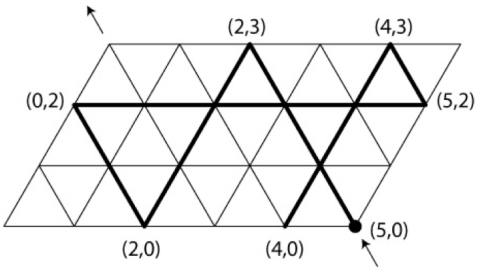
  We houden in het vervolg enkel rekening met efficiënte oplossingen.

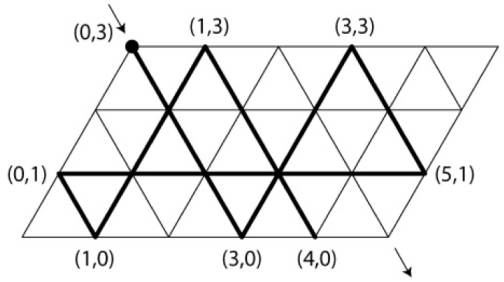
- De twee beschreven oplossingen zijn de enige. Bij de eerste stap heb je maar 2 mogelijkheden: ofwel vul je de 3 liter kan ofwel de 5 liter kan. daarna ligt elke stap vast door de efficiëntieregels.

- Kunnen we ook andere waarden afmeten met deze twee kannen. Dat we 3 en 5 liter kunnen afpassen is nogal duidelijk. Bij het eerste algoritme komen we bij stap 5 op (1,0), dus kunnen we ook 1 liter afpassen. Bij het tweede algoritme komen we bij stap 3 terecht op (0,2) en dus kunnen we ook 2 liter afmeten. met andere woorden: we kunnen elke inhoud van 1 to 5 liter, beide inhouden inbegrepen, afmeten met de twee gegeven kannen.

2.2 Grafische voorstelling

We kunnen de verschillende stappen ook grafisch voorstellen:





2.3 Op verkenning met andere inhouden.

- We nemen een kan van 3 liter en een kan van 7 liter. Ook nu gaan we na of we 4 liter kunnen afpassen.

- Als we starten met het vullen van de kleinste kan bekomen we volgend rijtje : $(0,0) \xrightarrow{2} (3,0) \xrightarrow{3} (0,3) \xrightarrow{2} (3,3) \xrightarrow{3} (0,6) \xrightarrow{2} (3,6) \xrightarrow{3} (2,7) \xrightarrow{1} (2,0) \xrightarrow{3} (0,2) \xrightarrow{2} (3,2) \xrightarrow{3} (0,5) \xrightarrow{2} (3,5) \xrightarrow{3} (1,7) \xrightarrow{1} (1,0) \xrightarrow{3} (0,1) \xrightarrow{2} (3,1) \xrightarrow{3} (0,4)$

- Het hele proces is gedetermineerd door de efficiëntieregels. We kunnen ook starten met de 7 lier kan en dan zijn we snel klaar: $(0,0) \xrightarrow{2} (0,7) \xrightarrow{3} (3,4) \xrightarrow{1} (0,4)$.

- Bij het eerste algoritme hebben we 6 keer de 3 liter kan gevuld en twee keer de 7 liter kan geledigd: $6 * 3 - 2 * 7 = 4$. Bij het tweede algoritme hebben we 1 keer de 7 liter kan gevuld en 1 keer de 3 liter kan leeg gegoten: $1 * 7 - 1 * 3 = 4$.

- Het is duidelijk uit de gegeven schema's dat elke inhoud kleiner dan of gelijk aan 7 verkregen kan worden met de 2 gegeven kannen.
- Het loopt echter niet altijd goed af! Neem maar eens een kan van 3 en een kan van 6 liter. je loopt vast: $(0,0) \xrightarrow{2} (3,0) \xrightarrow{3} (0,3) \xrightarrow{2} (3,3) \xrightarrow{3} (0,6) \xrightarrow{1} (0,0)$. Ook als je begint met de 6 liter kan geraak je er niet. Hoe komt dat? Als het verband klopt dat we gevonden hebben tussen enerzijds de gewenste hoeveelheid en anderzijds de inhouden van de gegeven kannen zou je 4 moet schrijven als een aantal keer 3 verminderd met een aantal keer 6 of omgekeerd: een aantal keer 6 verminderd met een aantal keer 3. Maar die verschillen zij altijd een drievoud en 4 is dat niet.

Hoofdstuk 3

Veralgemening

3.1 Probleemstelling

Veronderstel dat m, n, d natuurlijke getallen zijn verschillend van 0 en dat $0 < m < n$ en $d < n$.

> Het (d, m, n) probleem is de vraag of je met twee kannen van respectievelijk m en n liter inhoud een hoeveelheid van d liter kan afpassen?

De kannen hebben geen markeerstreepjes. Het (d, m, n) probleem is *oplosbaar* als er een algoritme kan worden gevonden om met de kannen van m en n liter juist d liter af te passen. In het vorige hoofdstuk hebben we vastgesteld dat het $(4, 3, 5)$ probleem oplosbaar is. Ook het $(4, 3, 7)$ probleem is oplosbaar. Het $(4, 3, 6)$ probleem is echter niet oplosbaar.

3.2 Oplosbaarheid

Het (d, m, n) probleem kan gemodelleerd worden door middel van de Diophantische vergelijking

$$mx + ny = d$$

waarvan de oplosbaarheid gegeven wordt door volgende stelling:

> Het (d, m, n) probleem is oplosbaar als en slechts als d een veelvoud is van de grootste gemene deler van m en n.

Zo is het $(4, 3, 5)$ probleem oplosbaar omdat ggd$(3, 5) = 1$ een deler is van 4.

3.3 Algoritmes

Veronderstellen we nu dat het (d, m, n) probleem oplosbaar is. Afhankelijk van welke kan we eerst vullen, zijn er twee mogelijke oplossingen bepaald door volgende algoritmes:

Algoritme M:

1. Definieer een dummy variabele k en zet die op 0.
2. Als $k \neq d$, blijf m toevoegen aan k en geef dat resultaat als nieuwe waarde van k totdat $k = d$ of $k > n$.
3. Als $k > n$ trek dan n af van k en ken dit resultaat toe aan k.
4. . Als $k = d$ stoppen we. Anders herhalen we stappen 2 tot 4.

Wat wordt dit bij het $(4, 3, 5)$ probleem?

- De algemene oplossing van de Diophantische vergelijking $3x + 5y = 4$ wordt gegeven door: $x = 3 + 5p$ en $y = -1 - 3p$
- Algoritme M geeft: $\boxed{0} \xrightarrow{3} \boxed{3} \xrightarrow{3} \boxed{6} \xrightarrow{-5} \boxed{1} \xrightarrow{3} \boxed{4}$
- Dit correspondeert met de oplossing $x = 3, y = -1$ voor $p = 0$ in de algemene oplossing van de Diophantische vergelijking. We moeten de kan van 3 liter drie keer vullen en de kan van 5 liter één keer leeggieten.
- Hiermee krijgen we volgende oplossing: $(0, 0) \to (3, 0) \to (0, 3) \to (3, 3) \to (1, 5) \to (1, 0) \to (0, 1) \to (3, 1) \to (0, 4)$.

We krijgen een ander algoritme als we de grootste kan beginnen te vullen.

Algoritme N:

1. Definieer een dummy variabele k en zet die op 0.
2. Als $k \neq d$, blijf n toevoegen aan k en geef dat resultaat als nieuwe waarde van k .
3. Als $k > d$ trek dan m af van k en ken dit resultaat toe aan k totdat $k = d$ of $k < m$.
4. . Als $k = d$ stoppen we. Anders herhalen we stappen 2 tot 4.

Wat wordt dit bij het $(4, 3, 5)$ probleem?

- De algemene oplossing van de Diophantische vergelijking $3x + 5y = 4$ wordt gegeven door: $x = 3 + 5p$ en $y = -1 - 3p$
- Algoritme N geeft: $\boxed{0} \xrightarrow{5} \boxed{5} \xrightarrow{-3} \boxed{2} \xrightarrow{+5} \boxed{7} \xrightarrow{-3} \boxed{4}$
- Dit correspondeert met de oplossing $x = -2, y = 2$ voor $p = -1$ in de algemene oplossing van de Diophantische vergelijking. We moeten de kan van 5 liter twee keer vullen en de kan van 3 liter twee keer leeggieten.
- Hiermee krijgen we volgende oplossing: $(0, 0) \to (0, 5) \to (3, 2) \to (0, 2) \to (2, 0) \to (2, 5) \to (3, 4) \to (0, 4)$.

3.4 Een voorbeeld

Kan je met een kan van 7 en en kan van 11 liter juist 6 liter afpassen?

- Het $(7, 11, 6)$ probleem is oplosbaar, want 7 en 11 zijn onderling ondeelbaar en hebben dus een grootste gemene deler gelijk aan 1; en 1 deelt 6.
- De overeenkomstige Diophantische vergelijking is $7x + 11y = 6$. De algemene oplossing is te schrijven al $x = -18 + 11k, y = 12 - 7k$. Met behulp van Python:

```
from sympy.solvers.diophantine import diophantine
from sympy import symbols
x, y = symbols("x, y", integer=True)
diophantine(7*x + 11*y - 6)
```

```
{(11*t_0 - 18, 12 - 7*t_0)}
```

- Als we algoritme M gebruiken, zoeken we de kleinst mogelijke positieve oplossing voor x en we vinden: $x = 4, y = -2$ voor $k = 2$. Dus 4 keer de kan van 7 liter vullen en 2 keer die van 11 liter leeggieten.
$\boxed{0} \xrightarrow{7} \boxed{7} \xrightarrow{7} \boxed{14} \xrightarrow{-11} \boxed{3} \xrightarrow{7} \boxed{10} \xrightarrow{7} \boxed{17} \xrightarrow{-11} \boxed{6}$

- Dit geeft het rijtje: $(0,0) \to (7,0) \to (7,7) \to (3,11) \to (3,0) \to (0,3) \to (7,3) \to (0,10) \to (7,10) \to (6,11) \to (6,0)$

- Volgen we algoritme N, dan zoeken we de kleinst mogelijke positieve oplossing voor y en we vinden dan $y = 5, x = -7$. We vullen de kan van 11 liter 5 keer en gieten die van 7 liter 7 keer leeg.
$\boxed{0} \xrightarrow{11} \boxed{11} \xrightarrow{-7} \boxed{4} \xrightarrow{11} \boxed{15} \xrightarrow{-7} \boxed{8} \xrightarrow{-7} \boxed{1} \xrightarrow{11} \boxed{12} \xrightarrow{-7} \boxed{5} \xrightarrow{11} \boxed{16} \xrightarrow{-7} \boxed{9} \xrightarrow{-7} \boxed{2} \xrightarrow{11} \boxed{13} \xrightarrow{-7} \boxed{6}$

- Dit geeft het rijtje: $(0,0) \to (0,11) \to (7,4) \to (0,4) \to (4,0) \to (4,11) \to (7,8) \to (0,8) \to (8,0) \to (1,0) \to (1,11) \to (7,5) \to (0,5) \to (5,0) \to (5,11) \to (7,9) \to (0,9) \to (9,0) \to (2,0) \to (2,11) \to (7,6) \to (0,6)$

3.5 Een oplossing met Python voor het (3,5,4) probleem

```
from collections import defaultdict

# kan1 en kan2 geeft de inhoud van de twee kannen
# en doel is de af te meten hoeveelheid.
kan1, kan2, doel = 3,5,4

# De bibliotheek krijgt beginwaarde False
bezocht = defaultdict(lambda: False)

# recursieve formule die de tussenstappen geeft
# om tot de oplossing te komen en die True of False weergeeft als het
# mogelijk is. a en b zijn de inhoeden van de kannen
#op een bepaaald moment.
def oplosser(a,b):


    if (a == doel and b == 0) or (b == doel and a == 0):
        print(a, b)
        return True

    # Controleert of je deze combinatie al hebt
    # Indien niet, gaat de procedure verder.
    if bezocht[(a,b)] == False:
        print(a, b)

        # Verandert de Boolse variabele als je de combinatie al had
        bezocht[(a,b)] = True

        # Controleert de 6 mogelijkheden
        # en ziet of er een kan
        return (oplosser(0, b) or
                oplosser(a, 0) or
                oplosser(kan1, b) or
                oplosser(a, kan2) or
                oplosser(a + min(b,(kan1-a)),
                b - min(b,(kan1-a))) or
                oplosser(a - min(a,(kan2-b)),
                b + min(a,(kan2-b))))

    # Geeft False als de combinatie reeds bezocht werd

    else:
        return False

print("Stappen: ")
oplosser(0,0)
```

```
Stappen:
0 0
3 0
3 5
0 5
3 2
0 2
2 0
2 5
3 4
0 4
```