Løsning til «Dyr som blir uvenner»

Dette er fortsettelsen på artikkelen «Dyr som blir uvenner» (Tangenten 1/2023). I sin generelle form lyder problemet «Dyr som blir uvenner» slik:

Vi har N par av dyr. Dyrene skal plasseres ved siden av hverandre i $2N$ stallbokser slik at

- mellom dyrene i par 1 er det ett annet dyr
- mellom dyrene i par 2 er det to andre dyr
- mellom dyrene i par 3 er det tre andre dyr
- osv.

Utfordringen er å finne en plassering som fungerer uansett hvor stor N er. Vi har allerede funnet ut at det finnes en slik generell løsning bare hvis N eller $N + 1$ er delelig med fire. Jeg har også beskrevet en søke-algoritme som finner en løsning ved å prøve, feile og justere. Det er ikke en generell løsning. En generell løsning angir plasseringen uten å prøve flere muligheter.

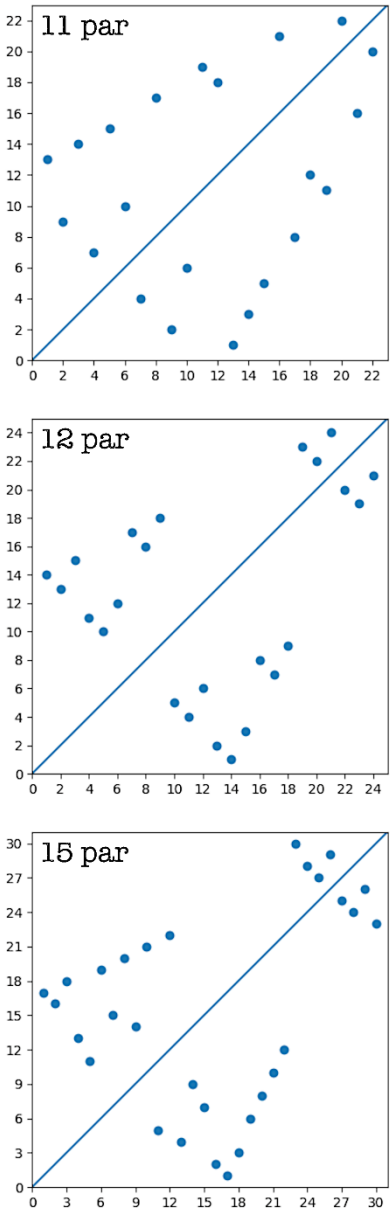
I det følgende vil jeg beskrive min vei til den generelle løsningen. Underveis oppstår det mange anledninger til matematisk utforskning som kan gjøres på mange ulike nivåer.

Avanserte undersøkelser

Søke-algoritmen som jeg har beskrevet i Tangenten 1/2023 begynner med paret med størst avstand. Hvert par plasserer jeg så langt til venstre som mulig. Så prøver jeg å plassere det neste paret. Hvis det ikke finnes plass til et par, må jeg gå tilbake til det forrige paret og korrigere dets plassering. Algoritmen finner en løsning etter at den har prøvd mange muligheter. Den er ganske ineffektiv og tar lang tid. En programmeringsoppgave til eldre elever kan være å optimalisere algoritmen.

Istedenfor å plassere parene i rekkefølgen fra størst til minst, kan vi velge paret med færrest muligheter. Det blir tydelig med fire par: Etter at vi har plassert parene 4 og 3, finnes det bare én mulighet for par 1: [4|**1**|3|**1**|0|4|3|0]. Og etterpå er det bare én mulighet for par 2 og problemet er løst: [4|1|3|1|**2**|4|3|**2**]. Dette går fortere enn å sjekke først alle mulige plasseringer for par 2. Med denne optimaliserte søke-algoritmen har jeg funnet mange løsninger opp til 100 par. Etterpå fant jeg nettsida til Miller (2020) som drøfter ulike måter man kan optimalisere søke-algoritmen på.

For å finne en generell løsning har jeg lett etter mønstre i løsningene, f.eks. grafisk.

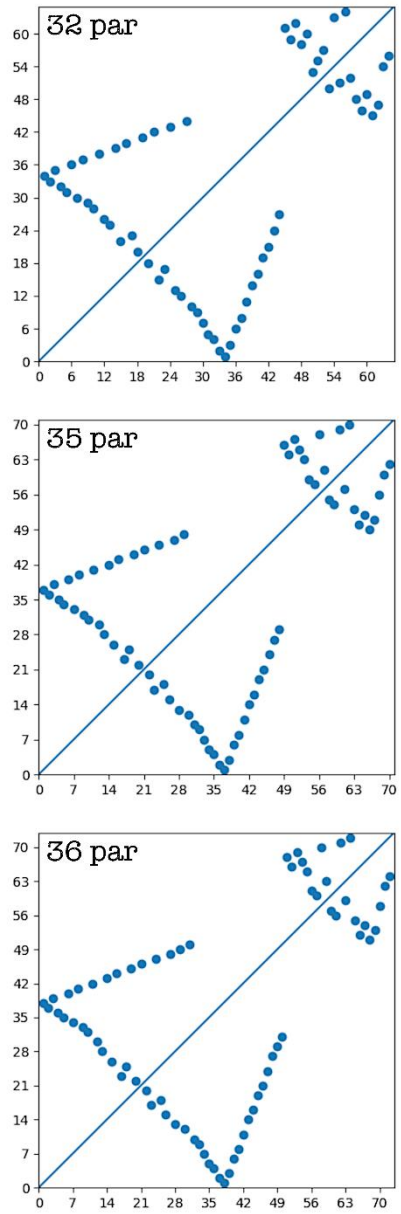


Figur 1. Løsninger for 11, 12 og 15 funnet med en søke-algoritme

Først har jeg plottet hvert dyr som punkt i et kartesisk koordinatsystem. Hvis et par er plassert i stallboksene p og q, får det ene dyret punkt (p, q) og det andre dyret punkt (q, p). Slik får jeg diagrammer som er speilsymmetriske med symmetriaksen $y = x$. De kan være ganske forskjellige, se for eksempel Figur 1. Jo flere par det blir, desto mer like blir de, se Figur 2. Likevel lyktes jeg ikke med å finne en generell løsning på denne måten.

En annen notasjon

Med min optimaliserte søke-algoritme blir alle løsningene mer like. For å undersøke likhetene har jeg valgt en annen fremstilling. Jeg har nummerert parene fra størst til minst, dvs. paret med størst mellomrom er $x = 0$, paret med nest størst mellomrom er $x = 1$, osv. og y-koordinaten er stallboksen til det venstre dyret. Det begynner også med null. Løsningen for fire par representeres altså av fire punkter: (0, 0), (1, 2), (2, 4) og (3, 1). I diagrammet oppdager vi interessante sammenhenger. Punktene ligger omtrent på linjer (se Figur 3). Videregående elever kan finne linjene ved å beregne en lineær regresjon. Allerede på tiende trinn kan elever tegne inn linjene og beregne stigningstallet.

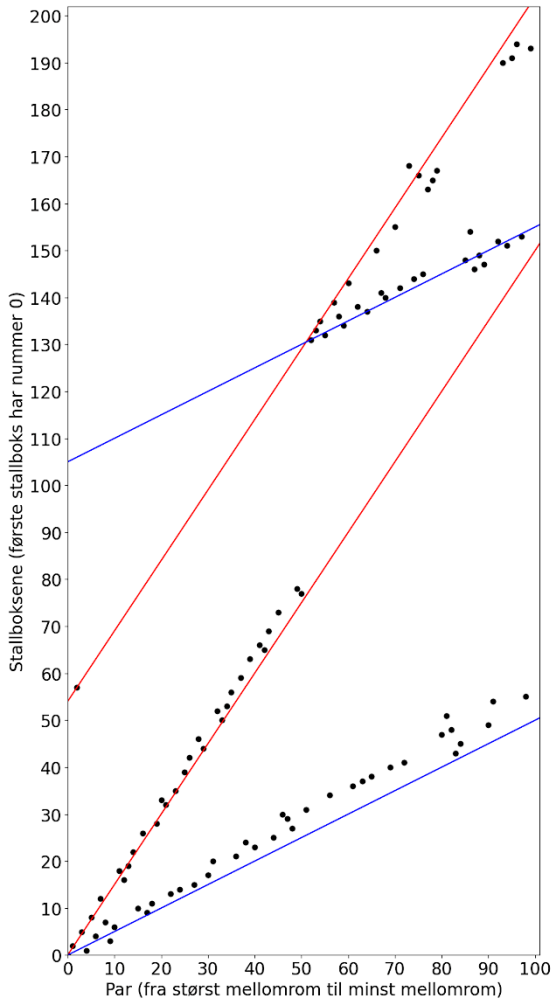


Figur 2. Løsninger for 32, 35 og 36 funnet med en søke-algoritme

Linjene i Figur 3 er ikke regresjonslinjer, men følger av disse resonnementene: De blå linjene har stigningstall 0,5. Det er par som er plassert «inni» hverandre. Da må nemlig avstanden minke med to når dyrene står direkte ved siden av hverandre, f.eks. [6|4|2|0|0|2|4|6]. Når x øker med to, øker y med én.

På de røde linjene ligger par som er plassert «etter» hverandre. Hvis parene plasseres direkte etter hverandre blir stigningstallet 2 fordi vi kan bruke bare annenhver stallboks, f.eks. [6|0|5|0|4|0|3|6|5|4|3]. Riktignok er stigningstallet i diagrammet ikke 2, men 1,5. Det får vi når vi plasserer annethvert par «etter» hverandre, f.eks. [6|0|0|4|0|0|2|6|4|2]. Når x øker med to, øker y med tre. Slik får vi kombinert begge deler: [6|**3**|**1**|**4**|**1**|**3**|**2**|6|**4**|**2**], dvs. punktene (**3, 1**), (**5, 2**) ligger på linjen $y = 0{,}5\,x - 0{,}5$, og punktene (**0, 0**), (**2, 3**), (**4, 6**) ligger på linjen $y = 1{,}5\,x$.

Det hadde vært en fin generell løsning, hvis vi kunne plassere alle partall «etter» hverandre og alle oddetall «inni» hverandre, men det fungerer dessverre ikke. I eksemplet mangler paret 5. Og i Figur 3 er det ganske tilfeldig hvilke par som følger hvilken linje. Derfor er det avvik fra linjene.



Figur 3. Plassering av 100 par. Parene ligger omtrent på linjer med stigningstall 0,5 eller 1,5.

Læreren som støttende stillas

De avanserte undersøkelsene trengs ikke for å finne den generelle løsningen. Elever på femte eller sjette trinn er i stand til å finne løsningen hvis de får hint fra læreren som kjenner løsningen. Hint som leder elevene på riktig spor uten å røpe løsningen kan være:

1. Prøv å plassere parene på en systematisk (regelmessig) måte.
2. Bruk papirstriper som hjelpemiddel.
3. Se først bare på antall par som er partall. For oddetallene finnes kanskje et annet mønster.
4. Bruk et eksempel som er så stort at systematikken blir synlig, men ikke altfor stort, f.eks. 12 eller 16.
5. Systematiske måter å plassere parene på er «etter» hverandre eller «inni» hverandre. Prøv å bruke én av disse fremgangsmåtene istedenfor å blande dem.

«etter»-strategien

Hvis vi har N par (med N delelig med fire), så er det mulig å plassere alle parene fra N til $\frac{N}{2}$ etter hverandre. For 12 par ser det slik ut:

[12|0|11|0|10|0|9|0|8|0|7|0|6|12|11|10|9|8|7|6|0|0|0|0].

Det er to problemer med det: På venstre side er alle mellomrommene mellom ledige stallbokser oddetall. Hvor skal vi plassere de resterende partallene? På høyre side er det $\frac{N}{2} - 2$ ledige stallbokser. Der kan parene $\frac{N}{2} - 4$ til 2 plasseres «inni» hverandre, men det finnes ikke plass til paret $\frac{N}{2} - 2$. Vi kunne frigjøre plass til dette paret ved å flytte noen par til venstre, dvs. fra «etter»-posisjonen til «inni»-posisjonen, men da følger vi ikke lenger en systematisk fremgangsmåte. Vi får det samme problemet som med søke-algoritmen: Hvilke par skal plasseres «etter» og hvilke «inni»?

Moore (1967) fant likevel en mulighet til å løse problemet med «etter»-strategien. Trikset var å begynne fra motsatt side. Slik er det også mye tydeligere at det handler om å plassere parene «etter» hverandre. Det er alltid mulig å plassere parene fra $\frac{N}{2}$ til N direkte etter hverandre hvis vi begynner fra venstre side. For 12 par ser det slik ut:

[6|7|8|9|10|11|12|6|0|7|0|8|0|9|0|10|0|11|0|12|0|0|0|0].

Nå kan vi plassere partallene mindre enn $\frac{N}{2}$ fra høyre ved å bruke annenhver stallboks.

[6|7|8|9|10|11|12|6|0|7|0|8|0|9|0|10|**4**|11|0|12|**2**|**4**|0|**2**].

Nå mangler det bare oddetallene som er mindre enn $\frac{N}{2}$. Alle stallbokser som fortsatt er ledige har mellomrom som er oddetall. Derfor vil vi alltid finne plasser for oddetallene, men vi må lete litt. For 12 par ser det slik ut

[6|7|8|9|10|11|12|6|5|7|1|8|1|9|5|10|4|11|3|12|2|4|3|2].

Strategien fungerer også hvis N er et oddetall, men da må vi begynne med $\frac{N+1}{2}$. Løsningen for 11 par ser slik ut:

[6|7|8|9|10|11|5|6|1|7|1|8|5|9|4|10|3|11|2|4|3|2].

Det finnes ikke en regel hvordan de siste oddetallene skal plasseres. Vi må til slutt prøve, feile og justere. Moore (1967) fant til og med flere muligheter hvis det er flere enn 20 par. Derfor er det ikke en generell løsning.

«inni»-strategien

Etter vi har plassert det største paret N (med N delelig med fire) helt til venstre har vi på venstre side (mellom dyrene) N ledige stallbokser og på høyre side $N-2$ ledige stallbokser. Vi kan plassere alle partall inni hverandre på venstre side og alle oddetall mindre enn $N-3$ på høyre side. Vi har to muligheter. Når vi bytter fra par- til oddetall, blir én stallboks ledig. Den kan være mellom par N og par $N-5$ eller helt på enden. For 12 par ser det slik ut:

[12|10|8|6|4|2|0|0|2|4|6|8|10|12
|0|7|5|3|1|0|1|3|5|7] eller
[12|10|8|6|4|2|0|0|2|4|6|8|10|12|
7|5|3|1|0|1|3|5|7|0].

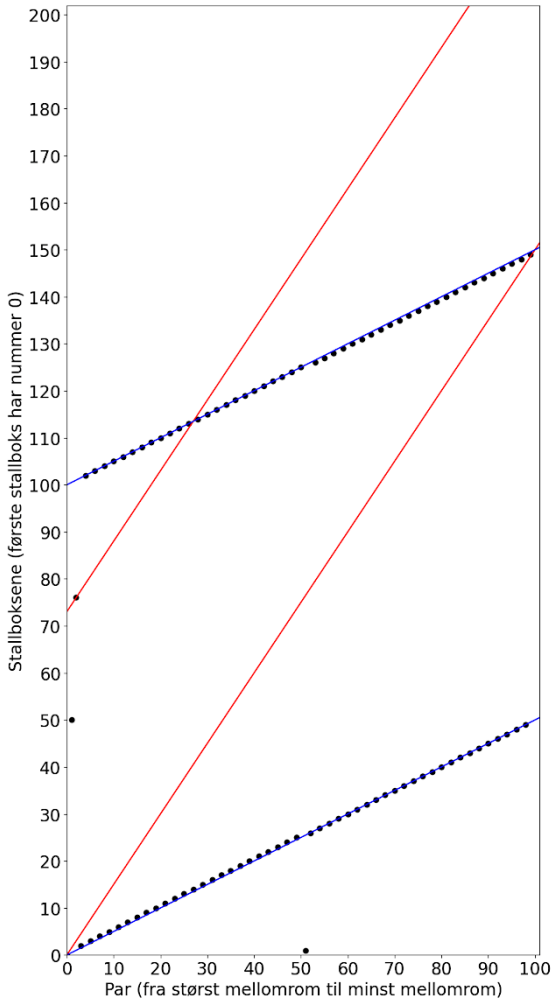
Uansett hva vi gjør, får vi ikke plassert parene $N-1$ og $N-3$.

Vi kan bytte par- og oddetall, dvs. vi plasserer alle oddetall mindre enn $N-1$ på venstre side og alle partall mindre enn $N-2$ på høyre side. Nå kan vi plassere $N-1$ eller $N-2$, men ikke begge deler:

[12|0|9|7|5|3|1|**11**|1|3|5|7|9|12|
8|6|4|2|0|**11**|2|4|6|8] eller
[12|0|9|7|5|3|1|**10**|1|3|5|7|9|12|
8|6|4|2|**10**|0|2|4|6|8].

Den generelle løsningen

Vi må blande par- og oddetall på hver side. For å fordele dem så jevnt som mulig, plasserer vi $\frac{N}{4}-1$ partall og like mange oddetall på hver side. På venstre side blir det oddetallene $N-3$ til $\frac{N}{2}+1$ og partallene $\frac{N}{2}-2$ til 2. På høyre side blir det partallene $N-4$ til $\frac{N}{2}$ og oddetallene $\frac{N}{2}-3$ til 1.



Figur 4: Den generelle løsningen for 100 par. Bortsett fra tre par ligger alle omtrent på linjer med stigningstall 0,5.

Ved overgangen fra par- til oddetall får vi én ledig stallboks. Den kan plasseres til venstre eller til høyre. Det skjer tre ganger: i overgangen fra N til $N-3$, i overgangen fra $\frac{N}{2}+1$ til $\frac{N}{2}-2$ og i overgangen fra $\frac{N}{2}$ til $\frac{N}{2}-3$. Det vil vise seg at det er best å plassere dem på plassene 2, $\frac{3}{4}N+2$ og $2N-(\frac{N}{4}-1)$. For 12 par ser det slik ut:

[12|0|9|7|4|2|0|0|2|4|0|7|9|12|8|6|3|1|0|1|3|0|6|8].

Slik får vi nemlig plassert de tre resterende parene: $\frac{N}{2}-1$ på plassene 2 og $\frac{N}{2}+2$ (mellom 2erne), $N-1$ på plassene $\frac{N}{2}+1$ (mellom 2erne) og $\frac{3}{2}N+1$ (mellom 1erne), og $N-2$ på plassene $\frac{3}{4}N+2$ og $2N-(\frac{N}{4}-1)$. Løsningen for 12 par ser slik ut:

[12|5|9|7|4|2|11|5|2|4|10|7|9|12|8|6|3|1|11|1|3|10|6|8].

Figur 4 viser løsningen for 100 par fremstilt på samme måten som i Figur 3. Bortsett fra parene $N-1$, $N-2$ og $\frac{N}{2}-1$ (som Figur 4 har i numrene 1, 2 og $\frac{N}{2}+1$) ligger alle par på eller nesten på de blå linjene. Hvis vi sammenligner figurene 3 og 4, blir forskjellen mellom algoritmene tydelig. Søke-algoritmen prøver å plassere hvert par så langt til venstre som mulig. Derfor er to tredeler av parene plassert i nedre halvdelen av diagrammet ($y < 100$). Den generelle løsningen fordeler parene jevnt. Omtrent annethvert par er plassert i øvre halvdelen av diagrammet (dvs. på den høyre sida av stallen).

Etter vi har løst problemet for N par med N delelig med fire, er det ganske lett å finne løsningen for $N-1$ par. Vi må fjerne paret N. Da får vi en ledig stallboks på plass $N+2$. Mellomrommet mellom den og paret $\frac{N}{2}-1$ er $2(\frac{N}{4}-1)+1=\frac{N}{2}-1$. Vi kan altså flytte dyret som sitter på plass 2 til plass $N+2$ og fjerne de to ledige stallboksene på venstre side. Slik har vi løsningen. For 11 par ser det slik ut:

1. [**0**|5|9|7|4|2|11|5|2|4|10|7|9|**0**|8|6|3|1|11|1|3|10|6|8]
2. [0|**0**|9|7|4|2|11|5|2|4|10|7|9|**5**|8|6|3|1|11|1|3|10|6|8]
3. [9|7|4|2|11|5|2|4|10|7|9|5|8|6|3|1|11|1|3|10|6|8]

Figur 5 viser en Python-funksjon som finner denne generelle løsningen for hvilket som helst tall.

```python
def løs_uvenner(antall_par: int) -> list:
    """Løser "Dyr som blir uvenner" for et gitt antall par

    Argument:
        antall_par (int): antall par som skal plasseres i stallen

    Returnerer:
        list: ei liste med lengde 2 * antall_par som inneholder alle dyrene på riktig plass
              eller ei tom liste hvis det ikke finnes en løsning
    """

    # Definere en hjelpefunksjon
    def rekke(plass: int, start: int, slutt: int):
        """Plasserer en rekke av par- eller oddetall inni hverandre i stallen.
           stall er en ikke lokal variable fra den overordnete funksjonen

        Argumenter:
            plass (int): plassen der rekken skal begynne
            start (int): tall som rekken skal begynne med
            slutt (int): tall som rekken skal slutte med
        """
        for tall in range(start, slutt-1, -2):
            stall[plass] = stall[plass + tall + 1] = tall
            plass += 1  # Fortsett med neste plass

    # Algoritmen fungerer bare hvis antall_par delelig med fire
    # eller antall_par + 1 er delelig med fire.
    if antall_par % 4 == 1 or antall_par % 4 == 2:
        return []  # ingen løsning

    # Hvis antall_par er et oddetall, løser vi først for antall_par + 1
    if antall_par % 2 == 1:
        stall = løs_uvenner(antall_par + 1)
        # Flytt dyret fra stallboks 2 (med indeks 1 i Python)
        # til stallboks antall_par + 2
        stall[antall_par + 1] = stall[1]
        # Returner stallen uten de første to stallboksene
        return stall[2:]

    # Lag en stall med 2 * antall_par ledige stallbokser
    stall = [0] * (2 * antall_par)

    halv = antall_par // 2  # Det brukes ofte
    kvart = halv // 2       # antall_par er delelig med fire

    # Vi begynner med det største tallet
    # Obs: den første stallboksen har indeks 0 i Python
    stall[0] = stall[antall_par + 1] = antall_par
    # Det neste er halvparten minus 1
    stall[1] = stall[halv + 1] = halv - 1

    # Store oddetall som går nedover, nesten til halvparten
    rekke(2, antall_par - 3, halv + 1)

    # Små partall som går nedover
    rekke(kvart + 1, halv - 2, 2)

    # Det nest største og det nest nest største tallet
    stall[halv] = stall[halv + antall_par] = antall_par - 1
    stall[halv + kvart + 1] = stall[halv + kvart + antall_par] = antall_par - 2

    # Store partall som går nedover
    rekke(antall_par + 2, antall_par - 4, halv)

    # Små oddetall som går nedover til 1
    rekke(antall_par + kvart + 1, halv - 3, 1)

    return stall
```

Figur 5. En Python-funksjon som beregner den generelle løsningen

Davies' løsning til Langfords problem

Allerede ett år etter Langford (1958) hadde stilt problemet, fant Davies (1959) en generell løsning. Han beviste at Langfords problem for N par kan løses bare hvis N er av formen $4m - 1$ eller $4m$ for alle naturlige tall m. Hans bevis ser annerledes ut enn mitt, men det handler om det samme. Hvis $N = 4m - 3$ eller $4m - 2$ finnes det en «krokete» løsning som har en ledig stallboks. Davies (1959) beviste at problemet kan løses ved å angi en generell løsning for $4m$ og $4m - 1$ par og en «krokete» generell løsning for $4m - 2$ og $4m - 3$ par. Hans løsninger for $4m$ og $4m - 1$ er de samme som mine, men med dyrene i omvendt rekkefølge. Han forklarer ikke hvorfor det er løsninger og beskriver heller ikke hvordan han har funnet løsningene.

I slutten av hans artikkel, lurer Davies (1959, s. 255) på om det finnes noen formel for hvor mange ulike løsninger det finnes for alle N. Det er et fortsatt uløst problem. Krajecki et al. (2016) brukte en ROMEO superdatamaskin med 500.000 GPU-kjerner. Den brukte 23 dager for å finne alle 1,6 trilliarder løsninger for 28 par. Langfords problem er et ganske populært kodingsproblem.

Konklusjon

Det er spennende å fordype seg i Langfords problem «Dyr som blir uvenner» uansett hvor gammel du er. Allerede barnehagebarn kan utforske det lekende. For skolebarn er det lett å variere vanskelighetsgraden bare ved å gi dem flere og flere dyr og etter hvert erstatte dyrene med tall. Og med litt hjelp er det mulig å finne en generell løsning. På vei mot den generelle løsningen vil elevene gjøre mange erfaringer med hvordan vi kan utforske et problem på en systematisk måte. De lærer også at det er verdifullt å prøve ut ting som ikke fører til en løsning. Underveis samler vi nemlig erfaringer som senere hjelper oss å forstå løsningen bedre.

Miller (2022) presenterer mange problemer som er beslektet med «Dyr som er uvenner», for eksempel å bruke trillinger eller firlinger istedenfor par, eller å bruke avstand istedenfor mellomrom. På hans nettside finner du også en litteraturliste og mye mer som henger sammen med problemet, for eksempel leker. Jeg er enig med Stephen Scattergood som sier: «Du lærer mye ved å prøve å løse dette problemet ... Ikke nødvendigvis så mye om selve problemet, men mye om de metodene du prøvde å bruke» (sitert etter Miller, 2022).

Referanser

Davies, R. O. (1959). On Langford's Problem (II). *The Mathematical Gazette, 43*(346), 253-255. https://doi.org/10.2307/3610650

Krajecki, M., Loiseau, J., Alin, F., & Jaillet, C. (2016). Many-Core Approaches to Combinatorial Problems: case of the Langford Problem. *Supercomputing Frontiers and Innovations, 3*(2), 21-37. https://doi.org/10.14529/jsfi160202

Langford, C. D. (1958). Problem. *The Mathematical Gazette, 42*(341), 228-228. https://doi.org/10.2307/3610395

Miller, J. E. (2020). *Notes on an Algorithm to Enumerate Langford Sequences*. Dialectrix. Hentet 26.12.2022 fra http://dialectrix.com/langford/langford-algorithm.html

Miller, J. E. (2022). *Langford's Problem*. Dialectrix. Hentet 26.12.2022 fra http://dialectrix.com/langford.html

Moore, D. (1967). *Constructing a Single Solution for any Valid n*. Dialectrix. Hentet 26.12.2022 fra http://dialectrix.com/langford/dave-moore.html

Forfatter

Oliver Thiel

Oliver.Thiel@dmmh.no

Dronning Mauds Minne