Tornen i Hanoi

Programmering II - Elixir

Kenan Dizdarevic

6 Februari 2023

Inledning

Tornen i Hanoi är ett matematiskt pussel som består av tre *pelare* och n antal skivor i olika storlekar. Skivorna kan endast placeras på så vis att ingen skiva som ligger ovanpå en annan får vara större än de som ligger under. Målet är att flytta alla skivor från en pelare till en annan, man får endast flytta en skiva i taget.

Pusslet verkar klurigt till en början. När man börjar prova sig fram inser man att det är lösbart. Men vi skall inte prova oss fram i vår lösning, vi skall lösa pusslet rekursivt i denna uppgift. Syftet är att få en bättre förståelse för rekursion.

Implementering

Rekursion

För att kunna lösa detta problem behöver vi påminna oss om vad rekursion är och hur det fungerar matematiskt. Rekursion är en matematisk metod där en funktion anropar sig själv för att lösa ett problem. Problemet löses genom att dela upp det i mindre beståndsdelar och lösa dem. Sedan slår man ihop dem och har en lösning för hela problemet. Det innebär att funktionen definierar sig själv i termerna av lösningen på en mindre version av samma problem. Vi kommer att komma till den nivån där vi har ett basfall som kan lösas utan rekursion.

Metoden för att lösa problem rekursivt presenteras nedan:

Låt $f(n)$ vara en rekursiv funktion.

1. Bevisa basfallet $f(1)$
2. Antag att $f(n)$ fungerar
3. Bevisa att $f(n+1)$ fungerar med $f(n)$

Vi skall nu studera hur man löser pusslet med tre skivor. Vi antar att vi har pelarna *vänster*, *mitten* och *höger*. De tre skivorna är placerade på vänstra pelaren när vi börjar, alla skall förflyttas till den högra pelaren i rätt ordning. Vi börjar med att flytta den minsta skivan till den högra pelaren, sedan flyttar vi nästa skiva till pelaren i mitten. Nu lägger vi den minsta skivan på pelaren i mitten och förflyttar den största skivan till pelaren till höger. Sedan förflyttar vi den minsta skivan till den vänstra pelaren samtidigt som vi flyttar skivan från pelaren i mitten till den högra pelaren. Slutligen flyttar vi den minsta skivan till den högra pelaren. Pusslet är löst!

Vi antar att $n = 3$ i detta fall. Vi flyttar $n - 1$ skivor till hjälppelaren, vi flyttar alltså två skivor till mitten från den vänstra pelaren. Sedan flyttar vi den sista skivan till den högra pelaren. Vår uppgift nu är att flytta de två skivorna från mitten till den högra pelaren. Tänker man efter inser man att vi har samma problem som vi hade i början. Nu är $n = 2$. Vi skall återigen flytta $n - 1$ skivor till hjälppelaren, vi flyttar således 1 skiva till den vänstra pelaren. Skivan i mitten flyttar vi till vår slutpelare, den högra pelaren. Avslutningsvis flyttar vi skivan från den vänstra pelaren till den högra pelaren.

Sekvensen

Eftersom beviset som vi presenterade för lösningen med tre skivor är giltigt, kan vi dra slutsatsen att det gäller för 4 skivor. Principerna är universella och de kan således tillämpas på n antal skivor. Vi har en generell lösning för pusslet.

Vi skall nu implementera funktionen `hanoi/4` som returnerar sekvensen av förflyttningarna för att lösa pusslet med n antal skivor.

Funktionen tar emot 4 argument, dessa är: *antal skivor*, *startpelaren*, *hjälppelaren*, *slutpelaren*.

Vi börjar med att implementera basfallet. I detta fall är det då vi inte har några skivor att förflytta. Vi returnerar således en tom lista. Koden är följande:

```
def hanoi(0, _, _, _) do [] end
```

Nu skall vi implementera det rekursiva fallet. Koden ser ut som följande:

```
def hanoi(n, startpelare, hjälppelare, slutpelare) do
    hanoi(n-1, startpelare, slutpelare, hjälppelare) ++
    [{:move, startpelare, slutpelare}] ++
    hanoi(n-1, hjälppelare, startpelare, slutpelare)
end
```

Det första steget i rekursionen är som vi tidigare nämnde är att flytta $n - 1$ skivor från början till hjälppelaren vilket är pelaren i mitten i detta fall. Detta görs med ett rekursivt anrop där vi specificerar att vi flyttar skivorna till hjälppelaren. Nästa steg är att skriva ut förflyttningen. Slutligen gör vi ett ytterligare rekursivt anrop där vi förflyttar skivorna från hjälppelaren till slutpelaren. I det sista rekursiva anropet börjar vi om med samma pussel, men med en skiva färre, och fortsätter på samma vis.

Vi testar om koden fungerar för 1, 2 och 3 stycken skivor, vilket den gör. Utskriften för att lösa pusslet med 4 skivor är följande:

```
[
  {:move, :startpelare, :hjälppelare},
  {:move, :startpelare, :slutpelare},
  {:move, :hjälppelare, :slutpelare},
  {:move, :startpelare, :hjälppelare},
  {:move, :slutpelare, :startpelare},
  {:move, :slutpelare, :hjälppelare},
  {:move, :startpelare, :hjälppelare},
  {:move, :startpelare, :slutpelare},
  {:move, :hjälppelare, :slutpelare},
  {:move, :hjälppelare, :startpelare},
  {:move, :slutpelare, :startpelare},
  {:move, :hjälppelare, :slutpelare},
  {:move, :startpelare, :hjälppelare},
  {:move, :startpelare, :slutpelare},
  {:move, :hjälppelare, :slutpelare}
]
```

Vi behöver genomföra 15 förflyttningar för att lösa pusslet med 4 skivor. Tabellen nedan presenterar förhållandet mellan antal skivor och antal förflyttningar.

Skivor	**Förflyttningar**
1	1
2	3
3	7
4	15

Tabell 1: Antal förflyttningar med avseende på antal skivor

För att hitta formlen som ger oss antalet förflyttningar med avseende på skivor behöver vi använda induktionsbevis.

1. Basfall: För $n = 1$ gäller formeln $2^n - 1$
2. Induktionsantagande: Antag att formeln $2^n - 1$ gäller för n stycken skivor
3. Induktionssteg: Bevisa att formeln gäller för $2^{n+1} - 1$, med $n+1$ skivor

Med $n + 1$ skivor flyttas de n första skivorna från pelaren vi börjar på till vår hjälppelare med $2^n - 1$ förflyttningar, enligt induktionsantagandet. Efter detta flyttas den sista skivan till slutpelaren med 1 förflyttning. Till sist förflyttas de n resterande skivorna till slutpelaren med $2^n - 1$ förflyttningar. Den matematiska beräkningen presenteras nedan:

$$2^n - 1 + 1 + 2^n - 1 = 2^{n+1} - 1$$

Q.E.D.

Med den bevisade formeln $f(n) = 2^n - 1$ kan vi nu beräkna antalet förflyttningar när pusslet har 10 skivor.

$$f(10) = 2^{10} - 1 = 1023$$

Det krävs således 1023 förflyttningar för att lösa pusslet med 10 skivor.

Slutsats

Slutligen inser vi att det är ett relativt simpelt problem om man har koll på rekursion och induktionsbevis.

Innan vi börjar med pusslet behöver vi systematiskt hitta en lösning. Vi försöker prova oss fram men det leder inte till en generell lösning. Vi ställer således upp ett induktionsbevis för att lösa problemet. Slutligen implementerar vi koden för lösningen.