Recursie

deel 2

Herhaling Hanoi

```python
def hanoi(n,a,b,c):
    if n > 0:
        hanoi(n-1,a,c,b)
        print(a,b)
        hanoi(n-1,c,b,a)

hanoi(5,1,2,3)
```

```
hanoi(3,1,2,3)
hanoi(2,1,3,2) + (1,2) + hanoi(2,3,2,1)
hanoi(1,1,2,3) + (1,3) + hanoi(1,2,3,1) + (1,2) + hanoi(2,3,2,1)
(1,2) + (1,3) + (2,3) + (1,2) + hanoi(2,3,2,1)
(1,2) + (1,3) + (2,3) + (1,2) + hanoi(1,3,1,2) + (3,2) + hanoi(1,1,2,3)
(1,2) + (1,3) + (2,3) + (1,2) + (3,1) + (3,2) + (1,2)

Let op: je moet vertrouwen op de recursie!
Deze uitschrijven maakt je meestal niet veel wijzer!
```

Aanpak Recursie

Probeer het probleem eentje kleiner te maken!

Dus:

- wat moet er nog gedaan worden als je er vanuit gaat dat het probleem voor 1 kleiner al is opgelost

Opgaven

Geef voor de volgende problemen een oplossing met een while -lus en een recursieve versie.

1. Maak in Python een functie find(x,xs) die als resultaat True heeft als het getal x in de lijst xs voorkomt en anders False.

2. Maak in Python een functie index(x,xs) die als resultaat de eerste positie van x in xs teruggeeft. Als x niet voorkomt moet het resultaat -1 zijn.

3. Maak in Python een functie isSorted(xs) die als resultaat True heeft als de lijst xs gesorteerd is van klein naar groot en anders False.

4. Maak in Python een functie count(x,xs) die als resultaat het aantal keren dat x voorkomt in xs.

5. Maak in Python een functie groep3(xs) die als resultaat de lijst xs in groepjes van 3 oplevert (een lijst van lijsten dus). Ga er vanuit dat het aantal elementen van xs een 3-voud is.

Subsets

Bepaal alle deelverzameling van een lijst:

Voorbeeld: [1,2,3], deelverzamelingen:

```
as [],[1],[2],[3],[1,2],[1,3],[2,3],[1,2,3]
```

Maak het probleem 1 kleiner, deelverzamelingen [2,3]:

```
bs [],[2],[3],[2,3]
```

Wat is het verschil tussen as en bs?

Subsets

```
as [],[1],[2],[3],[1,2],[1,3],[2,3],[1,2,3]
```

```
bs [],[2],[3],[2,3]
```

as bestaat uit de elementen van bs en nog een keer de elementen van bs met 1 aan het begin toegevoegd:

```
bs + [[1] + s for s in bs]
```

Wanneer stoppen we?

Als we de subsets van de lege lijst moeten bepalen:

`subsets([]) = [[]]` !!!!! Waarom niet `[]`?

Subsets

Python programma:

```
def subsets(k,xs):
    if k == len(xs):
        return [[]]
    else:
        h = a[k]
        ss = subsets(k+1,xs)
        return  ss + [[h] + s for s in ss]

a = [1,2,3,4]

print(subsets(0,a))
```

Opgave

subsets(xs) geeft alle deelverzamelingen van xs

choose(k,xs) geeft alle deelverzamelingen van lengte k

Geef een recursieve implementatie van choose(k,xs)

Let op: je moet zowel naar k als xs kijken!