

## algoritme

**cascadetrapp met**  $C_1, C_2, \dots, C_n$

$$C_1 \Rightarrow BF_1(C_2, C_3, \dots, C_n) \Rightarrow C'_1$$

$$C_2 \Rightarrow BF_2(C'_1, C_3, C_4, \dots, C_n) \Rightarrow C'_2$$

$$C_3 \Rightarrow BF_3(C'_1, C'_2, C_4, C_5, \dots, C_n) \Rightarrow C'_3$$

...

$$C_n \Rightarrow BF_n(C'_1, C'_2, \dots, C'_{n-1}) \Rightarrow C'_n$$

Bij  $BF_i$  als het antwoord “neen” is, dan weten we dat het sowieso niet in de bijhorende  $C_i$  zit.

Zo hebben we al veel elementen kunnen uitsluiten.

Dan de volgende cascade doen we met  $C'_1, C'_2, \dots, C'_n$

Het is wel duidelijk dat  $C'_1$  gemiddeld groter zal zijn dan  $C'_n$  en dezelfde volgorde opnieuw aanhouden zal deze kloof steeds meer en meer vergroten. De beste manier om dit tegen te gaan en daardoor minder cascadetrappen te hebben is om alles te sorteren op grootte en de grootste laatst te doen. Maar dit zal traag zijn, dus een mogelijkheid is om met een heuristiek te werken, bv. telkens omdraaien, eerst  $C_1, \dots, C_n$ , dan  $C'_n, \dots, C'_1$ , dan  $C''_1, \dots, C''_n$  enz.