

# VANDAAG

**SORTEREN**

**RECURSIE**

**OpenGL**

# **SORTEREN**

## **SORTEERMETHODEN**

**BY EXCHANGE**

**BY SELECTION**

**BY INSERTION**

## **IMPLEMENTATIE VAN SORTEERMETHODEN**

**ELEMENTEN VERPLAATSEN**

**POINTERS GEBRUIKEN**

## **EFFICIENTIE VAN SORTEERMETHODEN**

**AFHANKELIJK VAN AANTAL**

**AFHANKELIJK VAN GESORTEERDHEID**

# SORTEERMETHODEN

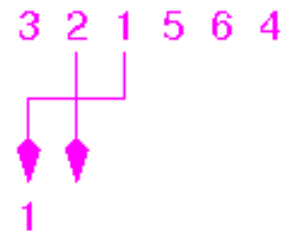
## BY EXCHANGE

ELEMENTEN PAARSGEWIJZE VERWISSELEN  
OM DE JUISTE ORDENING TE KRIJGEN



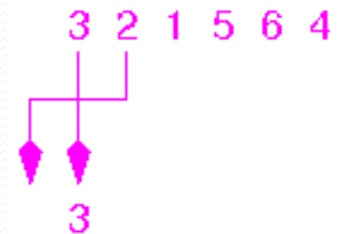
## BY SELECTION

ELEMENTEN IN DE JUISTE VOLGORDE  
ZOEKEN EN AAN EEN RIJ AL GESORTEERDE  
ELEMENTEN TOEVOEGEN



## BY INSERTION

ELEMENTEN NA ELKAAR OP DE JUISTE  
PLAATS TUSSENVOEGEN IN EEN RIJ AL  
GESORTEERDE ELEMENTEN



# IMPLEMENTATIE

**ELEMENTEN VERPLAATSSEN**

**VOOR SIMPELE ELEMENTEN IN ARRAYS  
EENVOUDIGER TE PROGRAMMEREN**

**POINTERS NAAR ELEMENTEN VERWISSELEN**

**VOOR GROTE ELEMENTEN  
VOOR COMPLEXE STRUCTUREN  
WERKT DAN VEEL SNELLER**

# EFFICIENTIE BIJ SORTEREN

**KAN ((ZEER) STERK) AFHANKELIJK ZIJN VAN:**

**HET SOORT ELEMENTEN**

**DE GESORTEERDHEID VAN DE ELEMENTEN**

**HET AANTAL ELEMENTEN**

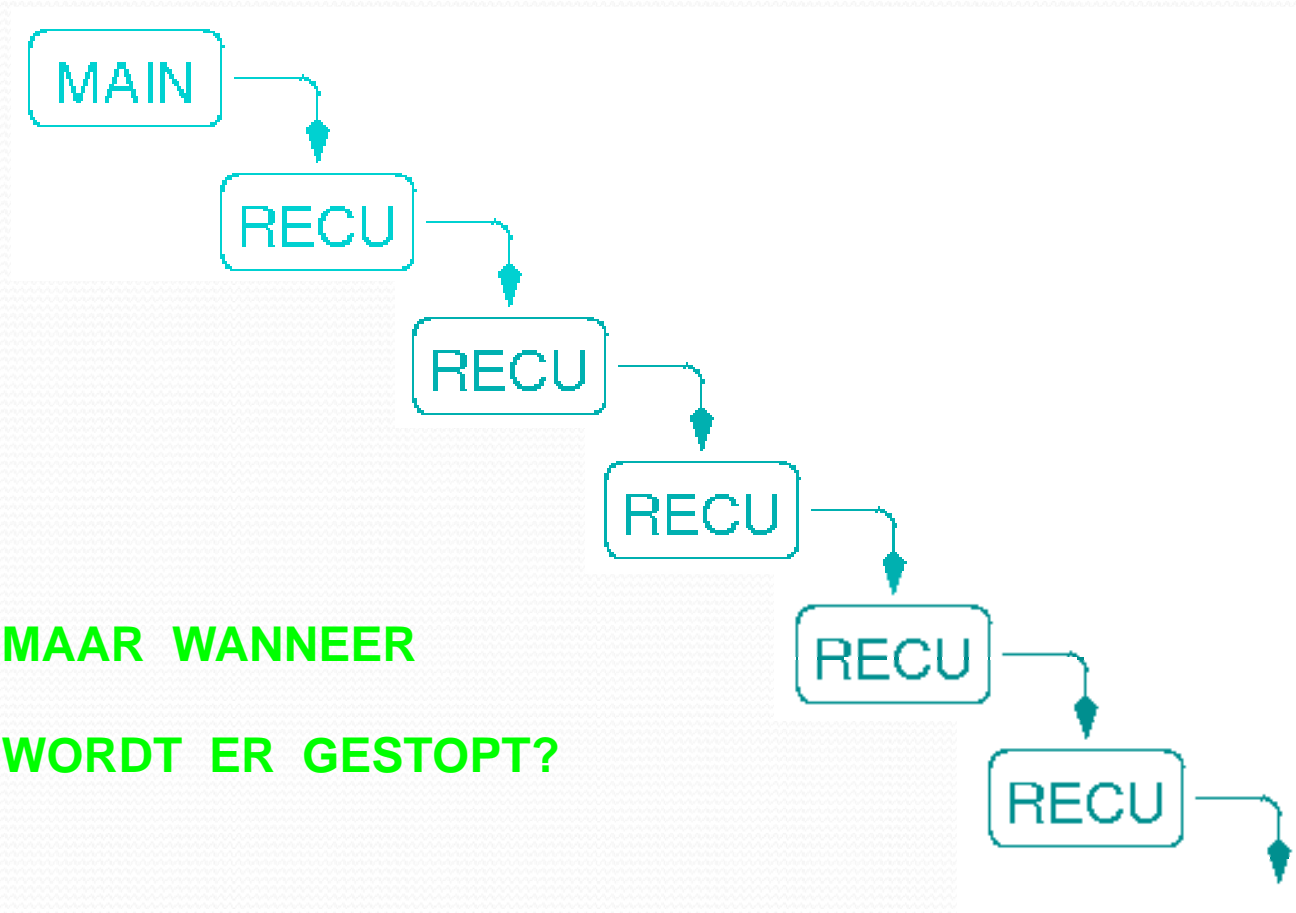
# SNELHEID SORTEERMETHODEN

SELECTION SORT	$\sim N^2$
INSERTION SORT	$\sim N^2$
BUBBLE SORT	$\sim N^2$
SHAKER SORT	$\sim N^2$
SHELL SORT	$\sim N^{1.2}$
QUICKSORT	$\sim N \log N$

DIT ZIJN GEMIDDELDE WAARDEN !

# WAT IS RECURSIEVE FUNCTIE?

EEN FUNCTIE DIE ZICHZELF AANROEPT



MAAR WANNEER

WORDT ER GESTOPT?

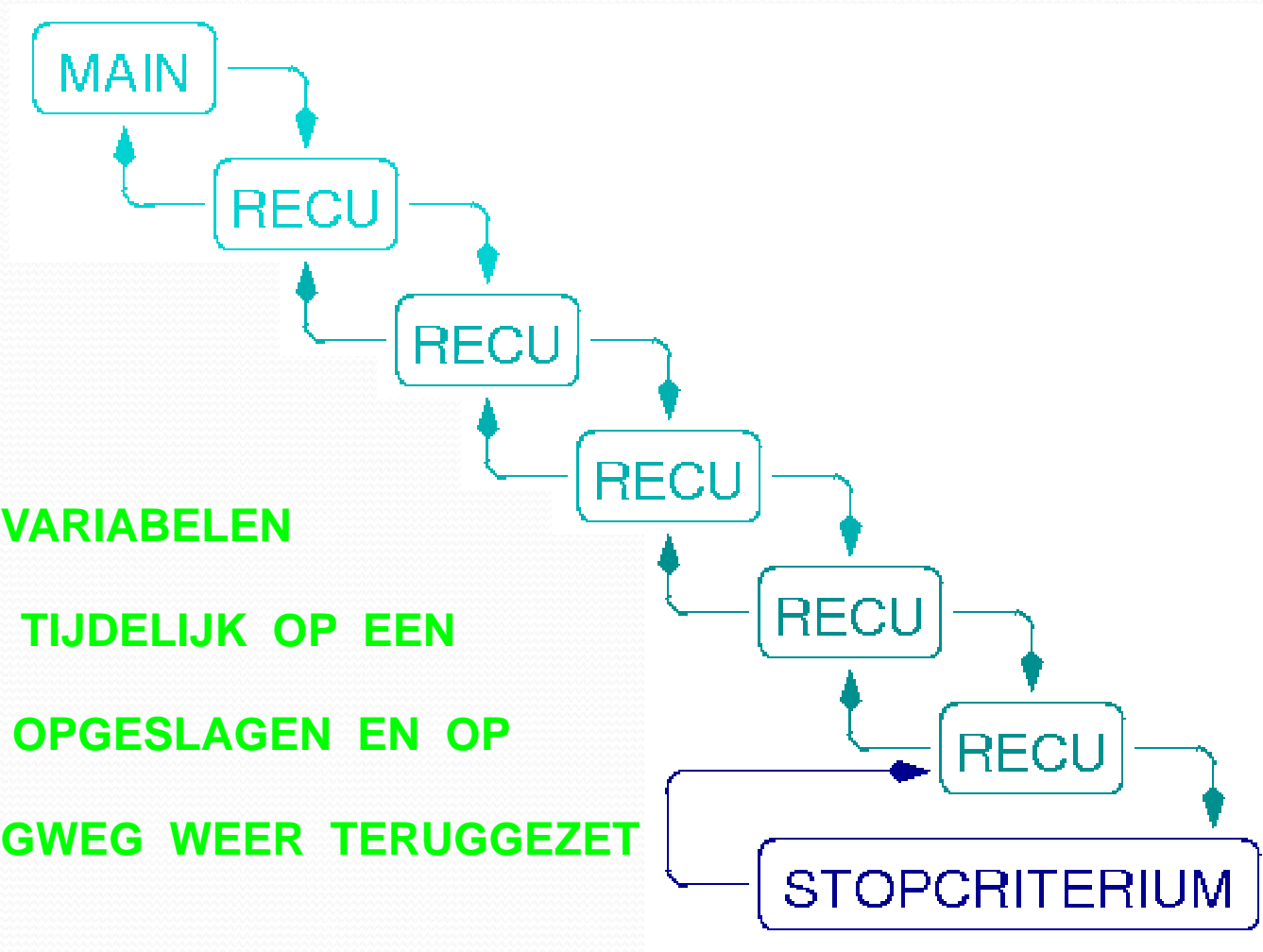
EN WAT GEBEURT ER INTUSSEN

MET DE WAARDEN VAN LOKALE

VARIABELEN VAN DE RECURSIEVE FUNCTIE?

# RECURSIE

EEN RECURSIEVE FUNCTIE BEVAT EEN TEST OM DE RECURSIVITEIT TE STOPPEN



LOKALE VARIABELEN

WORDEN TIJDELIJK OP EEN

“STACK” OPGESLAGEN EN OP

DE TERUGWEG WEER TERUGGEZET



# VOORBEELD ZONDER RECURSIE

```
double function X_tot_de_Nde_macht
(double x, int n)
// Bereken de n-de macht van x.
{
double product = 1.0;
int    macht;

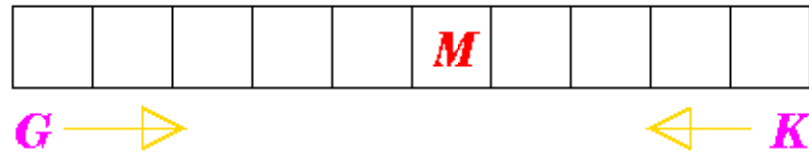
if (n>=0) // macht positief of nul
    for (macht=1; macht<=n; macht++)
        product *= x;
else // macht negatief
{
    for (macht=1; macht<=-n; macht++)
        product *= x;

    product = 1.0/product;
}
return (product);
}
```

# VOORBEELD MET RECURSIE

```
double function X_tot_de_Nde_macht
(double x, int n)
// Bereken de n-de macht van x.
{
if (n==0) // macht is 0
// Stoppen bij stopcriterium
return (1.0);
else
{
if (n>0) // macht is positief
// Doorgaan met kleinere n
return
(X_tot_de_Nde_macht(x,n-1)*x);
else // macht is negatief
// Doorgaan met positieve n
return
(1.0/X_tot_de_Nde_macht(x,-n));
}
}
```

# QUICKSORT



**M** = MEDIAAN, ANDERE ELEMENTEN WORDEN HIERMEE VERGELEKEN

**G** = ALLE ELEMENTEN LINKS HIERVAN ZIJN KLEINER DAN DE MEDIAAN

**K** = ALLE ELEMENTEN RECHTS HIERVAN ZIJN GROTER DAN DE MEDIAAN

## PROCEDURE

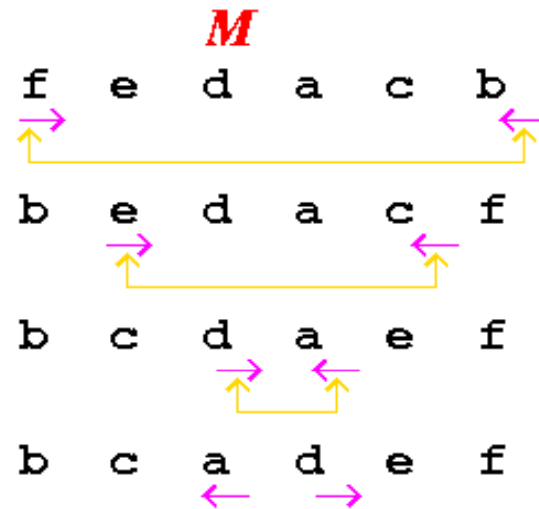
**G** ZOVER MOGELIJK NAAR RECHTS EN  
**K** ZOVER MOGELIJK NAAR LINKS  
OPSCHUIVEN

ALS **G** NIET VOORBIJ **K**, VERWISSEL DAN  
DE ERBIJ BEHORENDE ELEMENTEN,  
VERHOOG **G** MET 1, VERLAAG **K** MET 1

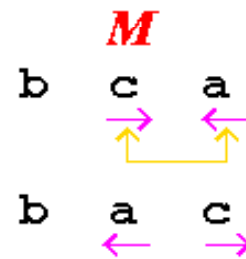
PROBEER DAARNA **G** EN **K** WEER ZOVER  
MOGELIJK OP TE SCHUIVEN

# QUICKSORT

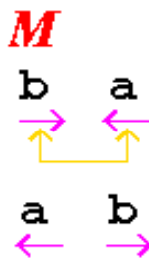
1E AANROEP  
REC. FUNCTIE



2E AANROEP



3E AANROEP





4E AANROEP

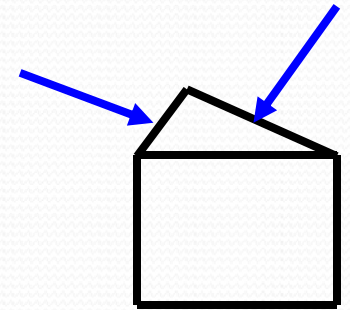


# Voorbeelden

- Recursie voorbeeld
- Simple sort versus Quick sort

# Algoritme voor het maken van de boom van Pythagoras

- Start met tekenen van basislijn
- • Herhaal onderstaande tot lengte basislijn  $<$  grenswaarde
  - Teken de drie resterende lijnstukken van het vierkant op de basislijn
  - Teken de twee resterende lijnstukken van de driehoek
  - Definieer als nieuwe basislijn opstaande zijde driehoek en roep functie aan bij 
  - Definieer als nieuwe basislijn andere opstaande zijde driehoek en roep functie aan bij 



# Voorbeelden

- Boom van Pythagoras
- Gebruik OpenGL