

## Complexiteit woordzoeken

Laat  $W$  een string van lengte  $M$  zijn en laat  $T$  een string van lengte  $N$  zijn met  $N \geq M \geq 1$  zijn. We gaan er vanuit dat de letters in  $W$  zijn opgeslagen in posities  $0 \dots M - 1$  en dat de letters in  $T$  zijn opgeslagen in posities  $0 \dots N - 1$ .

We willen het aantal voorkomens van het woord  $W$  in de tekst  $T$  tellen. Hiervoor kunnen we het volgende algoritme gebruiken:

```
teller = 0;
for i=0 to N-M do // i is mogelijke beginpositie van W in T
{ j=0;
  OK=true;
  while (j<M AND OK)
  { if (W[j] == T[i+j]) then
    j ++;
    else
      OK = false;
  }
  if (OK) then
    teller ++;
}
```

- a Hoeveel iteraties heeft de while-lus maximaal (in het slechtste geval dus) binnen één iteratie van de for-lus?
- b Druk de tijdscomplexiteit van het hele algoritme (voor het slechtste geval) uit in  $N$  en  $M$ .
- c Geef een voorbeeld van twee strings  $W$  en  $T$  waarvoor de hoeveelheid werk in dit algoritme maximaal is (een voorbeeld van het slechtste geval dus).