

Complexiteit heapsort

Ervanuitgaande

- dat er $N \geq 1$ getallen in het array A zijn opgeslagen, op posities $1 \dots N$

kunnen we heapsort op de volgende manier implementeren:

```
// maak een heap van het array A
for i=2 to N do
{ // laat A[i] omhoog borrelen
  j = i;
  while (j>=2 AND A[j/2]>A[j])
  { verwissel A[j/2] en A[j];
    j = j/2;
  }
}

// haal de getallen in oplopende volgorde uit de heap
TmpN = N;
for i=1 to N do
{ write A[1];
  A[1] = A[TmpN];
  TmpN --;

  // laat A[1] (de oude A[TmpN] dus) omlaag borrelen
  j = 1;
  OK = false;
  while (j<=(TmpN/2) AND !OK)
  { k = index van kleinste kind van A[j];
    // (k=2j of k=2j+1)
    if (A[j]>A[k])
    { verwissel A[j] en A[k];
      j = k;
    }
    else
      OK = true;
  }
}
```

- Hoeveel iteraties (afhankelijk van de waarde van i) heeft de while-lus in de eerste for-lus maximaal (in het slechtste geval dus) binnen één iteratie van de for-lus?
- Wat is de totale tijdscomplexiteit van de eerste for-lus (voor het slechtste geval)?
- Geef een voorbeeld van een rij getallen in A met $N = 10$ getallen, waarvoor de hoeveelheid werk in de eerste for-lus maximaal is (een voorbeeld van het slechtste geval dus).
- Hoeveel iteraties (afhankelijk van de waarde van TmpN) heeft de while-lus in de tweede for-lus maximaal (in het slechtste geval dus) binnen één iteratie van de for-lus?
- Wat is de totale tijdscomplexiteit van de tweede for-lus (voor het slechtste geval)?
- Geef een voorbeeld van een heap in A met $N = 10$ getallen, waarvoor de hoeveelheid werk in de tweede for-lus maximaal is (een voorbeeld van het slechtste geval dus).