

### Boblesortering uten stoppmerke

- den aller treigeste algoritmen, full gjennomgang hver gang
  - unødvendig, det største tallet fra hver gjennomgang kommer på riktig plass
  - og en trenger ikke sammenligne mot disse ved neste gjennomgang
- en kan unngå disse unødvendige sammenligningene ved å innføre stopp-merke og bare fortsette med gjennomganger hvis det ble bytte i siste gjennomførte gjennomgang
- [https://en.wikipedia.org/wiki/Bubble\\_sort](https://en.wikipedia.org/wiki/Bubble_sort)


### Boblesortering med stoppmerke

- den ytre løkka blir nå en WHILE-løkke, noe som skal hindre unødvendige gjennomganger
  - «WHILE-løkka avsluttes hvis det ikke blir bytter i forrige gjennomgang»
- i FOR-løkka som brukes til en gjennomgang «reduseres løkke-telleren med 1 for hver gang»

### Innstikksortering

- analogt/tilsvarende «kortstokk på hånda»
- noe raskere/mere effektiv enn boblesortering
  - ytre løkke er en FOR-løkke fra element nr 2 til siste element
- [https://en.wikipedia.org/wiki/Insertion\\_sort](https://en.wikipedia.org/wiki/Insertion_sort)

- Sortering - innstikk sortering - sammenlignings basert
- analogt / tilsvarende "kortstokk på hånda"
  - noe raskere / mer effektiv enn bubble sortering

 5 3 1 2 4 "1 bytte"

 3 5 1 2 4 "2 bytter"

 1 3 5 2 4 "2 bytter"

 1 2 3 5 4 "1 bytte"

1 2 3 4 5 sortert

- = "start på innstikket" / "neste kort"

2 koder for innstikkssortering med følgende/ulike  
metaforer for innstikk

5 3 1 2 4  
↓ ↓  
5 3 1 2 4

3 5 1 2 4  
↓ ↓ ↓  
3 5 1 2 4

1 3 5 2 4

"bytte hvert foran  
h1 finner riktig plass"

gjentakende "ta ut,  
sammenligne, sette  
foran"

5 3 1 2 4  
↓

- 3 tas ut ("gir plass" for  
forskyvning høyre)
- finner riktig plass og de  
andre forskyves høyre