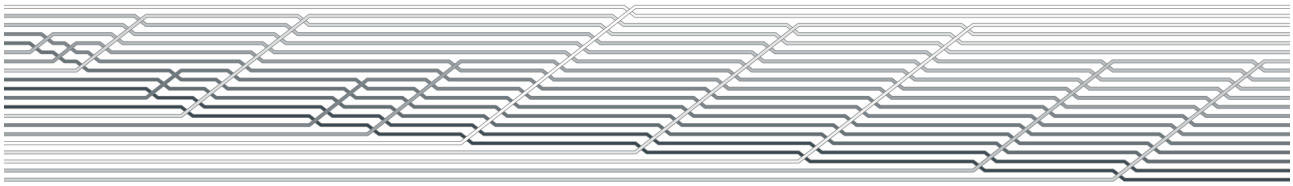
Gnome sort 10 poeng



Gnome sort er en enkel sorteringsalgoritme som *ikke* er kjent fra pensum. Illustrasjonen ovenfor er generert fra en kjøring av Gnome sort. Pseudokode for algoritmen er gitt nedenfor.

```
Input: Et array A med n elementer
Output: Et sortert array med de samme n elementene
1 Procedure GnomeSort(A)
2     i ← 0
3     while i < n do
4         if i > 0 and A[i-1] > A[i] then
5             A[i-1], A[i] ← A[i], A[i-1] // swap A[i] and A[i - 1]
6             i ← i − 1
7         else
8             i ← i + 1
```

I denne oppgaven skal du drøfte fordeler og ulemper ved Gnome sort sammenlignet med andre sorteringsalgoritmer kjent fra pensum. Teksten din bør inneholde:

- en kort forklaring på hvordan Gnome sort fungerer (med naturlig språk),
- kjøretidskompleksiteten til Gnome sort (sammen med en kort begrunnelse),
- en sammenligning av Gnome sort med sorteringsalgoritmer fra pensum, og
- en kort redegjørelse for hvorvidt Gnome sort er stabil og/eller in-place.

Gnome sort fungerer ved å løpe over elementene, og når den finner et element som er større enn det foregående elementet skal et flyttes bakover i arrayet. Oppførselen til Gnome sort er nesten helt lik Insertion sort, men den gjør dette med å kun vedlikeholde én indeks. Den mest vesentlige forskjellen fra Insertion sort er at den vil gjøre dobbelt så mange sammenligninger i det som svarer til én innsetting i Insertion sort.

Det er altså en sorteringsalgoritme med kvadratisk kjøretidskompleksitet (i likhet med bubble sort og selection sort). Den gjør akkurat like mange bytter som Insertion sort, men en god del flere sammenligninger.

På grunn av kjøretidskompleksiteten vil den være langt mindre egnet for store arrayer enn de raskeste sammenligningsbaserte sorteringsalgoritmene vi kjenner fra pensum (Merge, Heap og Quick).

Den er sammenligningsbasert, og er derfor vanskelig å sammenligne direkte mot Bucket- og Radix sort.

Algoritmen er (i likhet med Insertion sort) stabil, som er enkelt å se fordi den kun bytter elementer som står direkte ved siden av hverandre. Den er også in-place, fordi den ikke bruker andre datastrukturer enn arrayet som er gitt som input.

Dette er en litt friere oppgave, som gjør at sensor må yte eget skjønn. Her er noen veiledende kommentarer om poenggivning:

- Opptil 3 poeng for en god forklaring.
- Opptil 2 poeng for riktig kjøretidskompleksitet.
- Opptil 2 poeng for å «plassere» Gnome sort riktig i forhold til andre sorteringsalgoritmer.
- Opptil 2 poeng for en redegjørelse om stabilitet og in-place (disse krever lite forklaring).
- For full uttelling må studenten identifisere det svært nære beslektskapet med Insertion sort.

Det vektlegges at studenten viser god vurderingsevne med tanke på å trekke frem viktige momenter for å tegne et godt bilde av sorteringsalgoritmen sammenlignet med andre sorteringsalgoritmer. Lykkes de med det kan det gis noe mer uttelling enn hva punktene over foreslår.