

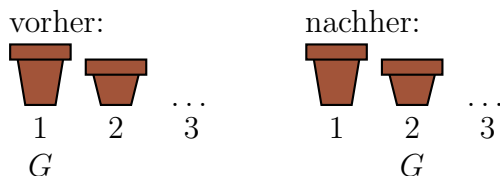
Worum geht es?

Gnomesort ist ein Sortieralgorithmus, der sehr einfach zu verstehen und zu programmieren ist. Er wurde von Hamid Sarbazi-Azad im Jahr 2000 unter dem Namen *Stupid Sort* veröffentlicht. Später wurde das Verfahren von Dick Grune *Gnome Sort* genannt.

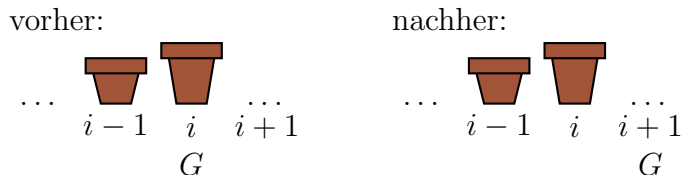
Der Algorithmus aus der Sicht eines Gartenzwergs

Ein Gnom G (*garden gnome*, Gartenzwerg) steht vor der Aufgabe, eine Reihe von n Blumentöpfen nach aufsteigender Grösse zu sortieren. Dabei führt er, abhängig von seiner Position, die folgenden Aktionen aus:

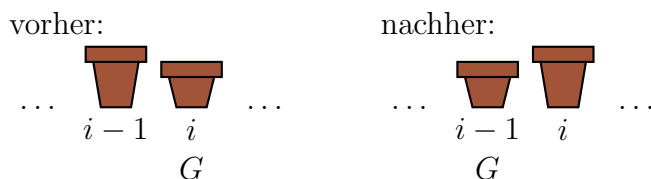
1. Steht er an Position 1, geht er einen Schritt nach rechts.



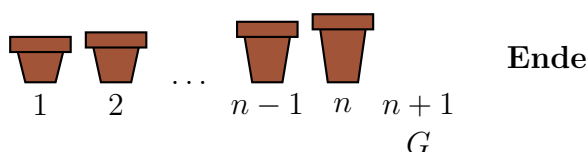
2. Steht er an der Position i ($1 < i \leq n$) und ist der linke Topf *nicht grösser* als der rechte, geht er eine Position nach rechts.



3. Steht er an der Position i ($1 < i \leq n$) und ist der linke Topf *grösser* als der rechte, vertauscht er beide Töpfe und geht danach eine Position nach links.



4. Steht er an der Position $n + 1$, ist er fertig und die Blumentöpfe sind aufsteigend sortiert.



Der Algorithmus in Pseudocode

```
GNOMESORT( $L$ )
1   $n \leftarrow L.length$ 
2   $i \leftarrow 1$ 
3  WHILE  $i \leq n$ 
4      IF  $i = 1$ 
5           $i \leftarrow i + 1$ 
6      ELSE IF  $L[i - 1] \leq L[i]$ 
7           $i \leftarrow i + 1$ 
8      ELSE:
9          vertausche  $L[i - 1]$  mit  $L[i]$ 
10          $i \leftarrow i - 1$ 
```

Aufgabe 1

Erstelle ein Python-Modul mit dem Namen `gnomesort.py` und implementiere darin die Funktion `gnomesort(L)`, welche die Liste `L` nach dem Gnomesort-Algorithmus sortiert.

- Beachte, dass im obigen Pseudocode das erste Element den Index 1 hat, während Python die Indizierung der Listen mit 0 beginnt.
- Will man in Python z.B. die Elemente `i=3` und `i=4` einer Liste `L` vertauschen, so geht das am einfachsten mit einer Mehrfachzuweisung: `L[3], L[4] = L[4], L[3]`

Aufgabe 1 (Lösung)

```
1 def gnomesort(L):
2     '''Sortiert die Elemente der Liste 'L' in place mit Gnomesort.'''
3     n = len(L)
4     i = 0
5     while i < n:
6         if i == 0:
7             i = i + 1
8         elif L[i-1] <= L[i]:
9             i = i + 1
10        else:
11            L[i], L[i-1] = L[i-1], L[i]
12            i = i - 1
13
14
15 L = [3, 8, 3, 2, 6, -1]
16 gnomesort(L)
17 print(L)
```

Aufgabe 2

Zeige schrittweise, wie Gnomesort die Liste rechts sortiert. Notiere den Zustand der Liste nach jeder Vertauschung.

7 2 3 9 5

Aufgabe 2 (Lösung)

7	2	3	9	5
---	---	---	---	---

2	7	3	9	5
---	---	---	---	---

2	3	7	9	5
---	---	---	---	---

2	3	7	5	9
---	---	---	---	---

2	3	5	7	9
---	---	---	---	---

Aufgabe 3

Wie viele Schritte (Aktionen) muss der Gartenzwerg in der folgenden Situation ausführen?



Aufgabe 3 (Lösung)

<u>1</u>	2	3	4	5
1	<u>2</u>	3	4	5
1	2	<u>3</u>	4	5
1	2	3	<u>4</u>	5
1	2	3	4	<u>5</u>
1	2	3	4	5

⇒ 5 Schritte

Aufgabe 4

Wie viele Schritte (Aktionen) sind nötig, um 1, 2, 3, 4, ... absteigend sortierte Blumentöpfe aufsteigend zu sortieren? Finde eine Formel für n Blumentöpfe



Aufgabe 4 (Lösung)

1 Blumentopf	2 Blumentöpfe	3 Blumentöpfe
<u>1</u>	<u>2</u> 1	<u>3</u> 2 1
1 _	2 <u>1</u>	3 <u>2</u> 1
	<u>1</u> 2	<u>2</u> 3 1
	1 <u>2</u>	2 <u>3</u> 1
	1 2 _	2 3 <u>1</u>
		2 <u>1</u> 3
		<u>1</u> 2 3
		1 <u>2</u> 3
		1 2 <u>3</u>
		1 2 3 _

Die Anzahl der Schritte beträgt 1, 4, 9, 16,

Vermutung: Die Anzahl der Schritte ist das Quadrat der Anzahl Blumentöpfe. Bei n Blumentöpfen sind somit n^2 Schritte nötig.