

Sortieralgorithmen - Selectionsort

Beim Sortieralgorithmus Selectionsort sind insgesamt so viele Schritte wie die dynamische Reihung lang ist - 1 zur Sortierung notwendig. Mit jedem Schritt wird ein weiterer Wert korrekt einsortiert (beim letzten Schritt werden zwei Werte einsortiert). Man sucht das Minimum der dynamischen Reihung und vertauscht die Position des Minimums mit der 0. Position der dynamischen Reihung. Anschließend ist die 0. Position sortiert und man sucht das Minimum zwischen der 1. Position der Liste und der letzten Position. Wieder wird die Position des Minimums mit dem Anfang des noch unsortierten Teils vertauscht.

Beispiel

4 2 1 6 3 5	Das Minimum ist 1. Vertausche also das 1. und das 3. Element.
1 2 4 6 3 5	Das Minimum des rechten Teilarrays ist 2. Da es bereits an 2. Position steht, wird es nicht getauscht.
1 2 4 6 3 5	Wir haben jetzt bereits ein sortiertes Teilarray der Länge 2. Wir vertauschen nun 4 und das Minimum 3.
1 2 3 6 4 5	Wir vertauschen 6 und 4.
1 2 3 4 6 5	Wir vertauschen 6 und 5.
1 2 3 4 5 6	Das Array ist jetzt fertig sortiert.

Wiederhole so oft, wie es Elemente in der dynamischen Reihung gibt - 1 und zähle i von 0
Suche das Minimum der dynamischen Reihung zwischen der i. und der letzten Position und merke die Position als k
Vertausche die i. Position mit der k. Position
Gebe die sortierte dynamische Reihung zurück

Funktion: finde die Position des Minimums eines Teils einer dynamischen Reihung
Speichere die erste zu betrachtende Position der dynamischen Reihung als min
Wiederhole von der ersten zu betrachtenden Position der dynamischen Reihung bis zum Ende der Reihung und zähle die Position k
Ist der Wert an der Position k kleiner als der Wert an der bisher gespeicherten Position min
Wahr
Falsch
Merke k als neues min

Funktion: tausche die Position zweier Elemente
speichere den Wert der ersten zu vertauschenden Position in x
speichere den Wert der zweiten zu vertauschenden Position in y
weise der dynamischen Reihung an der ersten Position den Wert y zu
weise der dynamischen Reihung an der zweiten Position den Wert x zu

Aufgabe 1

Gegeben sei eine dynamische Reihung, in welcher das Alter einer Gruppe von Personen enthalten ist. Diese wird z. B. durch den nachfolgenden Programmcode erzeugt.

```
alter = DynArray()  
alter.append(21)  
alter.append(20)  
alter.append(33)  
alter.append(30)  
alter.append(27)  
alter.append(28)  
alter.append(31)  
alter.append(19)
```

- a) Stellen Sie einen der anderen vorgestellten Sortieralgorithmen in schriftlich verbalisierter Form dar.
☞ **Hinweis:** Mergesort und Quicksort erfordern Rekursionen und sind daher deutlich schwieriger als Insertionsort und Bubblesort
- b) Überführen Sie den Algorithmus anschließend in ein Struktogramm und stellen Sie dieses dar.
- c) Implementieren Sie den oben dargestellten Sortieralgorithmus Selectionsort in Python.
- d) Implementieren Sie den Algorithmus anschließend in Python und testen Sie ihn an der dynamischen Reihung aus Aufgabe 2.