

LU01.A06 - Trace Table für lineare Suche



Erstellen Sie einen Trace Table für die lineare Suche in einer Liste, um den Ablauf des Algorithmus zu verstehen.

Erklärung der linearen Suche:

Die lineare Suche ist ein einfacher Suchalgorithmus, der ein Element in einer Liste durch sequentielles Überprüfen jedes Elements sucht, bis das gewünschte Element gefunden wird oder die Liste vollständig durchsucht wurde.

Anforderungen:

Schreiben Sie eine Funktion namens `linear_search`, die eine Liste von Ganzzahlen und ein zu suchendes Element als Parameter annimmt. Implementieren Sie den Algorithmus der linearen Suche. Erstellen Sie einen Trace Table, um die Änderung der Variablen während der Ausführung des Algorithmus nachzuvollziehen.

Beispielinput

```
zahlen = [3, 8, 2, 7, 5]
ziel = 7
```

Beispieloutput

```
Index des gesuchten Elements: 3
```

Lösungsvorlage

```
def linear_search(liste, ziel):
    for index, wert in enumerate(liste):
        if wert == ziel:
            return index
    return -1
zahlen = [3, 8, 2, 7, 5]
ziel = 7
resultat = linear_search(zahlen, ziel)
```

```
print(f'Index des gesuchten Elements: {resultat}')
```

Beispiel für den Trace Table:

Schritt	index	wert	Vergleich (wert == ziel)	Ergebnis (index oder -1)
1	0	3	Nein	-
2	1	8	Nein	-
3	2	2	Nein	-
4	3	7	Ja	3

Erklären Sie, was in jedem Schritt passiert und wie der Trace Table die Veränderung der Variablen zeigt, bis das gewünschte Element gefunden wird.

Eigenschaften der linearen Suche:



Die Suche ist einfach zu implementieren und funktioniert gut bei kleinen oder unsortierten Listen. Der Algorithmus durchsucht die Liste sequentiell, was im schlimmsten Fall $O(n)$ Vergleiche erfordert. Wenn das Element gefunden wird, wird der Index zurückgegeben, andernfalls -1.

From: <https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link: <https://wiki.bzz.ch/modul/m323/learningunits/lu01/aufgaben/tracetable1?rev=1722955963>

Last update: **2024/08/06 16:52**

