

Schreibe eine Methode, die ein Array der Größe n (n = Eingabeparameter) und den Werten 1 bis n belegt.

Die Werte sind dabei sortiert, siehe Beispiel:

charArray1

wert	1	2	3	4	5
Index	0	1	2	3	4

Schreibe eine Methode, die ein Array der Größe n (n = Eingabeparameter) und den Werten 1 bis n belegt.

Die Werte sind dabei zufällig sortiert. Nutze hierfür die [Random-Funktion](#), siehe Beispiel:

charArray2

wert	10	2	0	85	5
Index	0	1	2	3	4

Schreibe eine Methode namens `linSearch(int pEingabezahl)`, die überprüft, ob die eingegebene Zahl vorhanden ist.

Wie viele Durchläufe braucht die Methode, um herauszufinden, ob die Eingabezahl in dem sortierten Array vorhanden ist? Teste die verschiedene Szenarien der Tabelle und zähle die Anzahl der Durchläufe, indem du eine Zählvariable implementierst.

Größe des Arrays	Anzahl der Durchläufe		
	Suchzahl an Stelle 0	Suchzahl an Stelle $n/2$	Suchzahl an Stelle $n-1$
1			
10			
100			
1000			
10000			

Übertrage die Werte in das unten stehende Diagramm.

Wie viele Durchläufe braucht die Methode, um herauszufinden, ob die Eingabezahl in dem unsortierten Array vorhanden ist? Teste die verschiedene Szenarien der Tabelle und zähle die Anzahl der Durchläufe, indem du eine Zählvariable implementierst.

Größe des Arrays	Anzahl der Durchläufe		
	Suchzahl an Stelle 0	Suchzahl an Stelle $n/2$	Suchzahl an Stelle $n-1$
1			
10			
100			
1000			
10000			

Übertrage die Werte in das unten stehende Diagramm in einer anderen Farbe.

Diagramm Effizienzanalyse von linSearch:

