

# Den Grover-Algorithmus verstehen

## Arbeitsblatt

Ziel ist es, den Grundgedanken und die Funktionsweise des Grover-Algorithmus zu erfassen, insbesondere wie er die Suche in einer unsortierten Liste von Elementen beschleunigen kann. Für diese Aufgabe müsst ihr zu fünft sein.

### Schritt 1: Vorbereitung

- Erstellt eine Liste mit vier Elementen (dies können Zahlen, Buchstaben oder Objekte sein, z. B. A, B, C und D).
- Wählt eines der Elemente aus. Es ist das „versteckte Element“, das gefunden werden soll, beispielsweise das Element C.

### Schritt 2: Einen klassischen Suchvorgang simulieren

- Einer von euch spielt eine Person, die auf einem klassischen Computer nach einem versteckten Element sucht. Die anderen vier werden den vier Elementen zugewiesen (ohne dass die suchende Person weiß, wer welches Element repräsentiert).
- Der klassische Computer kann immer nur ein Element auf einmal überprüfen. Die suchende Person macht eine Abfrage, d. h. sie fragt einen der vier Mitschüler\*innen: „Bist du das richtige Element?“
- Der klassische Computer antwortet entweder mit „Ja“ oder „Nein“. Notiert euch, wie viele Abfragen erforderlich sind, bis das richtige Element gefunden ist.
- Erläutert den **günstigsten Fall** (das Element wird bei der ersten Abfrage gefunden), den **durchschnittlichen Fall** (zwei Abfragen sind erforderlich) und den **schlechtesten Fall** (drei Abfragen sind erforderlich, da das richtige Element das letzte in der Liste ist).

*Tipp:* Bei  $N$  Elementen beträgt die durchschnittliche Anzahl der Abfragen bei einem klassischen Suchvorgang  $N/2$ . In unserem Fall ist  $N = 4$ , was bedeutet, dass die durchschnittliche Anzahl der Abfragen  $4/2 = 2$  beträgt.

**Wie viele Abfragen sind erforderlich, um ein verstecktes Element in einer unsortierten Liste mit vier Elementen ( $N = 4$ ) zu finden?**

Die möglichen Fälle sind:

**Fall 1 – ja nein nein nein** → erfordert 1 Abfrage = die erste Schülerin ist das versteckte Element (ja), die drei Schüler\*innen, die neben ihr stehen, sind es nicht (nein).

**Fall 2 – nein ja nein nein** → erfordert 2 Abfragen.

**Fall 3 – nein nein ja nein** → erfordert 3 Abfragen.

**Fall 4 – nein nein nein ja** → es ist keine zusätzliche Abfrage erforderlich: Wenn man weiß, dass das dritte Element nein ist, weiß man, dass das vierte Element ja ist.

Da alle Positionen gleich wahrscheinlich sind, beträgt die durchschnittliche Anzahl der Abfragen  $N/2 = 2$ .

### Schritt 3: Mit dem Grover-Algorithmus einen Suchvorgang simulieren

1. Vier Schüler\*innen werden den vier Gegenständen A, B, C und D zugeordnet, wobei eine\*r davon das versteckte Element ist. Zeichnet zwei Linien auf den Boden, schreibt „Ja“ an die eine und „Nein“ an die andere. Die vier Schüler\*innen stellen sich irgendwo zwischen den beiden Linien auf. Sie alle haben die gleiche Wahrscheinlichkeit, die richtige Antwort (= das versteckte Element) zu sein.
2. Die „suchende Person“ fragt immer wieder das Orakel:<sup>1</sup> „Wo befindet sich das versteckte Element?“ Bei jeder Abfrage macht die Schülerin, die das versteckte Element darstellt, einen kleinen Schritt in Richtung der „Ja“-Linie (nur minimal, sodass die suchende Person noch nicht ganz sicher sein kann, dass sie das richtige Element ist). Die Schülerin wird „markiert“. Gleichzeitig machen die anderen drei Schüler\*innen einen kleinen Schritt in Richtung der „Nein“-Linie. So funktioniert der Grover-Algorithmus – er markiert das Element, ohne es konkret zu überprüfen.
3. Weitere Verstärkung:  
Wenn Schritt 2 mehrmals wiederholt wird, bewegt sich die Schülerin, die das versteckte Element darstellt, immer mehr in Richtung der „Ja“-Linie, während sich die anderen in Richtung der „Nein“-Linie bewegen. Dadurch erhöht sich die Wahrscheinlichkeit, dass die markierte Schülerin die richtige Antwort ist, bzw. verringert sich die Wahrscheinlichkeit, dass die anderen Schüler\*innen die richtige Antwort sind. Mit jeder Wiederholung tritt die „versteckte“ Schülerin deutlicher hervor.
4. Messung:  
Nach nur zwei dieser „Markieren und Verstärken“-Zyklen werdet ihr wahrscheinlich erkennen, wer das „markierte“ Element ist, da diese Person am nächsten an der „Ja“-Linie steht. Sie ist die endgültige „ermittelte Antwort“.

### Schritt 4: Vergleich mit dem klassischen Suchvorgang

Nimmt man nur vier Elemente, wird der tatsächliche Vorteil gegenüber einer klassischen Suche nicht besonders deutlich. Es zeigt nur die Funktionsweise des Grover-Algorithmus. Der Grover-Algorithmus verringert die Anzahl der Abfragen, die nötig sind, um ein verstecktes Element unter  $N$  Elementen zu finden, auf ungefähr  $\sqrt{N}$ .

Der Vorteil des Grover-Algorithmus wird aber sehr schnell deutlich, wenn die Anzahl der Elemente größer wird. Deshalb solltet ihr diese Übung vielleicht mit einer größeren Gruppe von Schüler\*innen wiederholen, beispielsweise mit 16. In diesem Fall würde eine klassische Suche durchschnittlich  $16/2 = 8$  Abfragen erfordern, während der Grover-Algorithmus in der Regel etwa  $\sqrt{16} = 4$  Abfragen benötigt.

---

<sup>1</sup> Ein „Orakel befragen“ ist eine Methode beim Programmieren. Man kann damit zum Beispiel eine Entscheidungsfrage beantworten (etwa „dieses oder jenes?“). Der Name bezieht sich auf ein Orakel in der griechischen Antike – eine Person oder ein Gegenstand, die/das Antworten auf Fragen zur Zukunft gab oder Orientierungshilfe für eine zu treffende Entscheidung bot.

## Schritt 5: Diskussion & Fragen

1. Wie unterscheidet sich das Orakel von einem klassischen „Hinweis“?
2. Was bewirkt der Verstärkungsschritt?
3. Beschreibt den günstigsten, den schlechtesten und den durchschnittlichen Fall bei der Verwendung einer klassischen Suche, um einen Gegenstand unter 12 unsortierten Gegenständen zu finden.
4. Überprüft, ob die durchschnittliche Anzahl der Abfragen  $6 (= N/2)$  beträgt, mit der im obigen Kasten beschriebenen Methode.
5. Wie viele Schritte wären erforderlich, um den Gegenstand mit dem Grover-Algorithmus zu finden?

