ANLEITUNG

Divide & Conquer

Divide and Conquer (deutsch Teile-und-herrsche-Verfahren) ist eine in der Informatik gängige Methode zur Entwicklung von Algorithmen. Solche Algorithmen zerlegen ein Problem rekursiv in Teilprobleme, bis sie einfach genug sind, um direkt gelöst zu werden. Rekursive Problemlösung bedeutet hier, dass eine Funktion programmiert wird, die sich aus ihrem eigenen Code heraus aufrufen lässt. Das wird anhand des folgenden Beispiels sehr viel klarer.

Divide & Conquer im Detail

Zur Veranschaulichung von Divide and Conquer (D&C) betrachtet nochmal das Problem aus dem Video »Die Zauberbohne«. Es sollten aus 1024 Bohnen die eine Bohne gefunden werden, die etwas leichter ist als die anderen. Divide and Conquer wird wie folgt angewandt:

- Teilt das Problem in ähnliche Unterprobleme (»Divide«), d. h. teilt den Stapel in zwei gleichgroße Stapel.
- Jetzt lautet das Unterproblem »Welche Stapelhälfte ist leichter?«
- Dann löst Ihr das Teilproblem rekursiv, d. h. Ihr findet die Lösung, indem Ihr eine kleinere Version desselben Problems löst. Also findet die leichtere Stapelhälfte, indem Ihr die leichtere Hälfte der Stapelhälfte findet 🤯 (»Conquer«)
- Das macht Ihr so oft, bis Ihr den »Basisfall« erreicht, der so einfach ist, dass er direkt gelöst werden kann. Bei der Bohnensuche ist der Basisfall der Vergleich von Stapeln, die jeweils nur aus einer Bohne bestehen, von denen eine die gesuchte Bohne ist.

Beim Zauberbohnen-Such-Beispiel seid Ihr an diesem Punkt bereits fertig. Bei anderen D&C-Algorithmen gibt es noch einen letzten weiteren Schritt, nämlich:

- Kombiniert die Lösungen der Unterprobleme zur Hauptlösung (»Combine«).

Dies wird zum Beispiel vom D&C-Sortieralgorithmus »Merge Sort« verwendet: Zuvor sortierte kleinere Listen (Teilprobleme) werden so kombiniert, dass man am Ende eine sortierte Liste mit allen Einträgen erhält.

Die Macht von Divide & Conquer

Divide and Conquer ist ein sehr mächtiges Werkzeug, das eine Menge Zeit sparen kann. Am Beispiel der Bohnensuche zeigt sich, dass im schlimmsten Fall statt 1024 naiver Vergleiche nur zehn nötig waren. Wenn es viermal so viele Bohnen gewesen wären, wären nur zwei weitere Vergleiche erforderlich gewesen, und bei achtmal so vielen Bohnen wären nur drei weitere Vergleiche nötig gewesen. Betrachtet man andere Fälle, so lässt sich ein Muster erkennen, das lautet:

$$\textit{Anzahl der Vergleiche} = \log_2(\textit{Anzahl der Bohnen})$$

Das heißt, die Zahl der Vergleiche mit D&C wird nie größer sein als die Zahl, die sich ergibt, wenn man den Logarithmus zur Basis zwei der Bohnenanzahl nimmt.
Keine Bange! Ihr müsst dieses Muster nicht selbst erkannt haben. Es dient eher als Ausblick für diejenigen von Euch, die in Zukunft tiefer in die Welt der Programmierung eintauchen wollen. Die Formel beschreibt bloß, dass selbst wenn Ihr nach sehr, sehr vielen Artikeln sortieren oder suchen wollt, die Anzahl der erforderlichen Vergleiche kaum zunimmt. Und mehr braucht Ihr davon nicht mitnehmen.

Vielen Dank!
Wir freuen uns
über Euer Feedback.