

# Übungsblatt Algorithmen - Rekursion, Backtracking

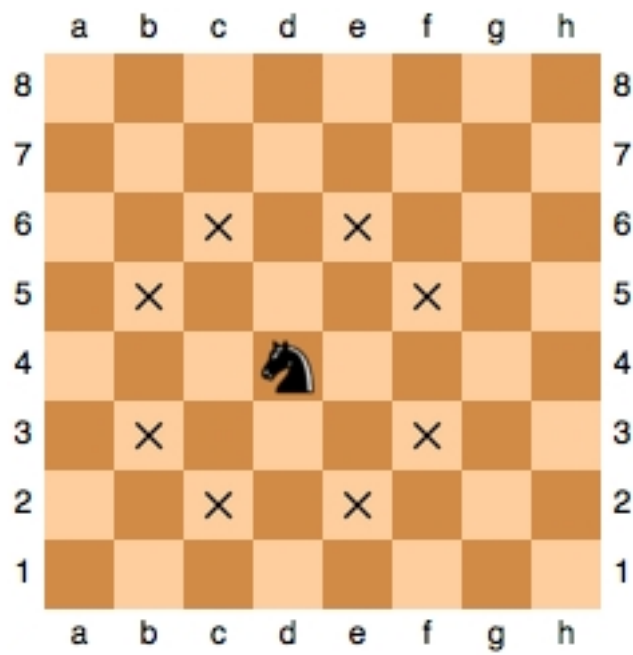
## Aufgabe 1

In der Vorlesung wurde ein rekursiver Algorithmus zur binären Suche vorgestellt. Erstellen Sie einen Algorithmus, der das binäre Suchen iterativ löst.

## Aufgabe 2

Entwerfen Sie einen Algorithmus für das Springerproblem.

Im Original wird das Springerproblem auf einem Schachbrett, also auf einem Brett der Größe  $8 \times 8$  gelöst. In einer Verallgemeinerung sind alle Brettgrößen  $m \times n$  möglich. Der Springer startet auf einem beliebigen Feld und muss durch eine erlaubte Zugfolge (siehe Abb.) alle Felder des Brettes besuchen. Wenn er zusätzlich am Anfangsfeld wieder ankommt, so heißt die Zugfolge geschlossen.



Überlegen Sie sich, wie groß die Schachtelungstiefe bei einem rekursiven Algorithmus, der Backtracking verwendet, maximal wird. Lohnt sich eine rekursive Umsetzung?

Man kann ausgehend von einer Situation alle möglichen Nachfolgesituationen in einem Baumdiagramm darstellen. Wie viele Zugmöglichkeiten hat man in einer Situation? Schätzen Sie daraus grob nach oben ab, wie viele Knoten (und damit wie viele Zugfolgen) sich für ein  $m \times m$  Feld ergeben.