

User Handbuch für die Webseite Minimax- Simulator

Diese Webseite bietet einen Simulator für die folgenden Algorithmen:

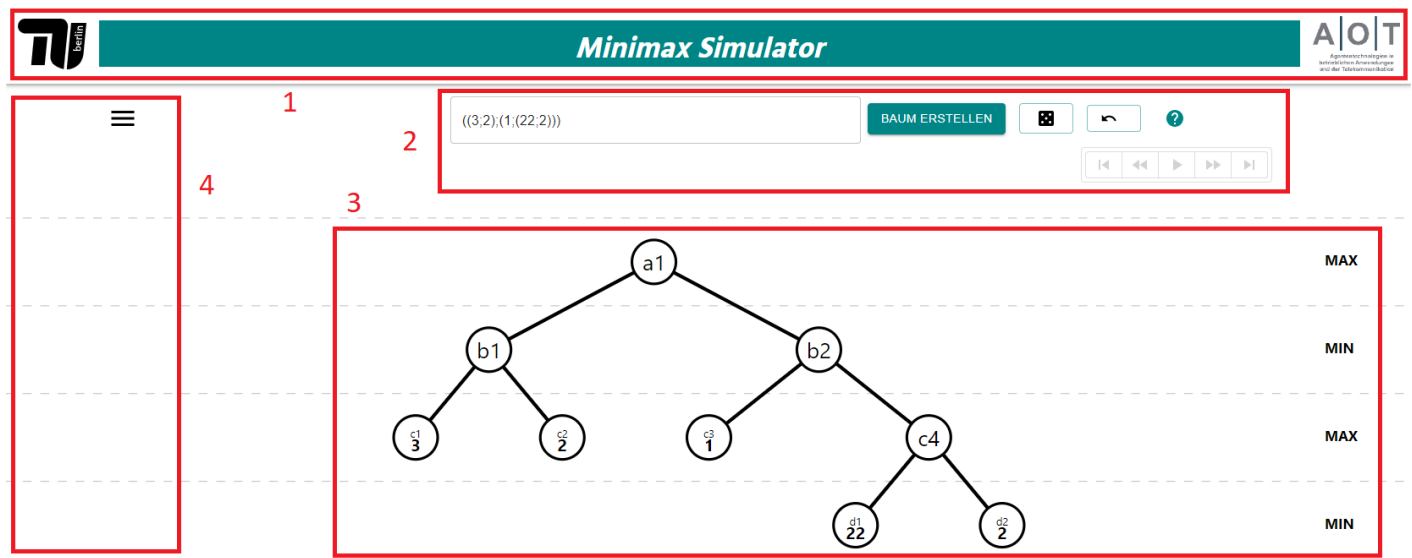
- Minimax
- Alpha-Beta
- IDDFS (Iterative Tiefensuche)

Die Webseite ist hauptsächlich auf zwei Zielgruppen ausgerichtet:

1. Der Kursleiter benötigt ein Tool, um Spielbäume zu erstellen, die dann im Übungsbetrieb oder auch in Klausuren verwendet werden können.
2. Den Studierenden hilft dieses Werkzeug, die Algorithmen durch Schritt-für-Schritt-Simulationen besser zu verstehen.

Beschreibung der Webseite:

Die Webseite besteht aus 4 Hauptkomponenten:



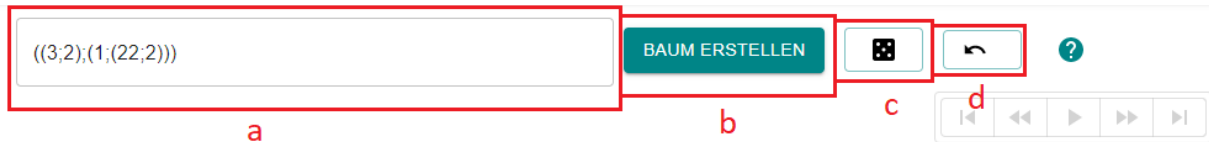
1. Header:

In der Kopfzeile sind drei Elemente zu sehen: Das TU-Berlin-Logo, das AOT-Logo und ein Titel. Die zwei Logos sind statisch, der Titel passt sich an *den momentan betrachteten Algorithmus* an.

Wird beispielsweise der IDDFS-Algorithmus ausgewählt, steht in der Kopfzeile "Iterative Tiefensuche Simulator".

2. Eingabefelder:

Auf der Höhe des Eingabefeldes befinden sich vier Komponenten.



- a. **Textfeld:** Hier kann ein Baum als String eingegeben werden.

Kodierung von Spielbäumen

Wir kodieren Spielbäume durch Angabe des **Wurzelknotens**. Ein Knoten wird dabei wie folgt kodiert:

`WERT(Kind1;Kind2;...;KindN)`

wobei WERT der **Wert** des Knotens ist. Falls der Knoten keinen Wert hat, wird das Feld weggelassen. Die **Kindknoten** werden rekursiv als Knoten kodiert. Bei einem **Blattknoten** müssen die Klammern für die Angabe der Kindknoten weggelassen werden. Der leere String wird als wertloser Wurzelknoten interpretiert.

Es folgt ein Beispiel. Der Spielbaum

spielbaum

wird kodiert durch

`(-2;(7;((1);0));(-9))`

Durch Klicken auf das Eingabefeld lassen sich die schon erzeugten Bäume anzeigen und wiederherstellen.

- b. **Baum Erstellen:** Nach Eingabe eines Strings kann mittels des Buttons "Baum erstellen" ein neuer Baum angezeigt werden. Das Baum-String-Format lässt sich am besten verinnerlichen, indem man ein paar Bäume auf diese Weise erstellt. Mit der Enterfunktion lässt sich der Baum gespeichert.
- Bei einer gültigen Eingabe wird sowohl der Graph angezeigt als auch eine kleine grüne Mitteilung unten mit der Nachricht "Baum erfolgreich erstellt."
 - Bei einer ungültigen Eingabe, z.B. aufgrund einer fehlenden Klammer, wird nur eine Fehlermeldung mit der entsprechenden Nachricht angezeigt.

- c. **Würfel:** Hierbei handelt es sich um einen "Auf gut Glück"-Button. Er erstellt einen zufälligen Baum basierend auf bestimmten Parametern. Im Default-Fall beträgt die minimale Baumtiefe 3, die maximale Tiefe 4, der Verzweigungsgrad 4 und die Wahrscheinlichkeit, einen Kindknoten zu erzeugen, 70%. Diese Parameter lassen sich jedoch in einem gewissen, die Funktionstüchtigkeit der Webseite sicherstellenden Rahmen im Reiter "Bearbeiten" des Seitenmenü ändern.
- d. **Undo-Pfeil:** Bei einem Klick wird die letzte Änderung am Baum rückgängig gemacht.

3. Graph:

Das Herzstück der Seite bildet der große, mittig platzierte Graph, auf dem die Algorithmen ausgeführt werden.

Die gestrichelten Linien trennen die Ebenen des Baumes. Am Rand ist vermerkt, ob der maximierende oder der minimierende Spieler in dieser Ebene am Zug ist.

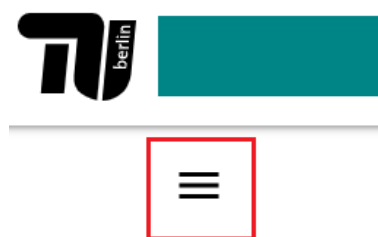
Die Namen jedes Knoten folgen diesem Schema:

- Die Wurzel wird initial mit dem Namen "a1" bezeichnet.
- Der Buchstabe im Namen zeigt die Tiefe der Ebene an: a für Tiefe 0, b für Tiefe 1 et
- Die Zahl im Namen gibt an, um den wievielten Knoten es sich von links nach rechts gelesen handelt.
- Der Name "d7" beschreibt also einen Knoten der Tiefe 4, der genau 6 weitere Knoten links von sich auf derselben Ebene hat.







4. Seitenmenü:

Standardmäßig ist das Seitenmenü zugeklappt.

Bei einem Klick auf den Seitenmenü-Button lässt sich das Seitenmenü öffnen.




Das Seitenmenü lässt sich mit einem Klick auf den Pfeil-Button, der sich oben rechts vom Seitenmenü befindet, wieder zuklappen.

		
	Datei	a
	Simulieren	b
	Analyse	c
	Bearbeiten	d
	Anzeige	e

Das Seitenmenü besteht aus vier Reitern:

a. Datei:

 Datei

Importieren

DATEI HOCHLADEN

Exportieren

Dateiformat

.svg

SPEICHERN

SPEICHERN UNTER...

SIMULATION SPEICHERN

Unter Importieren lässt sich eine Datei im .txt-Format hochladen sowie abspeichern. Die Datei beinhaltet den Baumstring, die Statistiken zu den Algorithmen und die Simulationsschritte. Man kann ebenso auswählen, ob sowohl die Statistik als auch die Simulationsschritten in der Datei gespeichert werden soll.

Ein Beispiel für eine Datei ist:

Baumstring:

$((3;2);(1;(22;2)))$

Statistiken:

Baumdaten:

Knotenanzahl: 26
Baumtiefe: 4

Minimax:
Besuchte Knoten: 26
Besuchte Blätter: 26

Alpha-Beta $[-\infty, \infty]$:
Besuchte Knoten: 19
Besuchte Blätter: 9
Cutoffs: 3

Alpha-Beta [4, 39]:
Besuchte Knoten: 17
Besuchte Blätter: 7
Cutoffs: 5


IDDFS:
Besuchte Knoten: | Tiefe 0: 1 | Tiefe 1: 3 | Tiefe 2: 6 | Tiefe 3: 13 | Tiefe 4: 24 |
Besuchte Blätter: | Tiefe 0: 0 | Tiefe 1: 0 | Tiefe 2: 0 | Tiefe 3: 3 | Tiefe 4: 12 |

IDDFS mit [4, 39]:
Besuchte Knoten: | Tiefe 0: 1 | Tiefe 1: 3 | Tiefe 2: 5 | Tiefe 3: 11 | Tiefe 4: 17 |
Besuchte Kanten: | Tiefe 0: 0 | Tiefe 1: 0 | Tiefe 2: 0 | Tiefe 3: 1 | Tiefe 4: 7 |
Cutoffs: | Tiefe 0: 0 | Tiefe 1: 0 | Tiefe 2: 2 | Tiefe 3: 2 | Tiefe 4: 4 |

Simulationsschritte: . . .

Unter Exportieren besteht zusätzlich die Möglichkeit, den Baum als .svg und png.

b.Simulieren:

 Simulieren

Algorithmus:






Algorithmus
MiniMax


Alpha Fenster

Beta Fenster

SIMULATION STARTEN

Steuerung:





Unter Algorithmus gibt es folgende Auswahlmöglichkeiten:

- MiniMax
- Alpha-Beta
- IDDFS

Bei den letzteren zwei besteht die Möglichkeit, die Alpha- und Beta-Fenster zu nutzen. Der Default ist $-\infty$, $+\infty$. Je nach Alpha- und Beta-Eingaben, dem aktuellen Baum und dem ausgewählten Algorithmus werden die Schritte des Algorithmus simuliert.

Unter Steuerung befinden sich "Player Buttons" mit den Optionen:

- Zum Anfang springen (Pfeiltaste nach unten) Ein Schritt zurück (Pfeiltaste nach links) Automatisch laufen lassen
- Ein Schritt vorwärts (Pfeiltaste nach rechts) Zum Ende springen (auch durch die Pfeiltaste nach oben bedienbar).

Die Simulationsleiste lässt sich außerdem durch ein zusätzliches Fenster auch bedienen, wenn das Seitenmenü geschlossen wird.

Anhand der Tastenfunktionen kann man die Simulation ein Schritt vorwärts bzw. rückwärts fahren lassen. Beim Klicken der Leertaste kann die Simulation angefangen bzw. gestoppt werden.


Unter den "Player Buttons" gibt es einen Slider, mit dem die Simulationsschritte schneller durchlaufen werden können.

Wenn der IDDFS-Algorithmus ausgewählt ist, gibt es zusätzlich die Buttons "Prev" und "Next", mit denen zwischen den verschiedenen Iterationstiefen gewechselt werden kann. Bei wiederholtem Bedienen von "Next" erreicht man schließlich den iterativen Durchlauf, in dem die Tiefe der Gesamtbaumtiefe entspricht.

c. Analyse:

Hier werden sämtliche relevanten Informationen und Statistiken zu dem angezeigten Baum und den drei Algorithmen bezogen auf dem Baum dargestellt: So z.B. Knotenanzahl, Baumtiefe, die besuchte Knoten, abgeschnittene Knoten bzw. Blätter und Anzahl Cutoffs.

d. Bearbeiten:

 Bearbeiten

Knoten Werkzeuge:

HINZUFÜGEN

EINFÜGEN

LÖSCHEN

AUTO-BENENNUNG

Zweig Werkzeuge:

TAUSCHEN

LÖSCHEN

Zufälliger Baum Optionen:

Minimale Tiefe
2

Maximale Tiefe
4

Maximale Verzweigung
3

Zufall
0.5

Hinweis:

Während der Simulation ist die Bearbeitung des Baumes nicht möglich. Beenden Sie die Simulation um den Baum zu bearbeiten.

Je nach gewünschter Aktion können im Baum ein oder zwei Knoten markiert werden. Unter Knoten-Werkzeuge kann man auf diesen Knoten folgende Aktionen ausführen:

- **Hinzufügen:** Erstellt ein Kind ausgehend vom markierten Knoten.
- **Einfügen:** Erstellt einen neuen Parent-Knoten ausgehend vom markierten Knoten.
- **Löschen:** Den markierten Knoten löschen.
- Hinweis: Beim Löschen eines Knoten bleiben die Kinder im Baum.
- **Auto-Benennung:** Durch das Hinzufügen von Knoten gerät die alte Benennung der Knoten durcheinander. Das kann durch Betätigen dieses Buttons korrigiert werden.


Unter Zweig-Werkzeuge können folgende Aktionen vorgenommen werden:

- **Tauschen:** Zwei markierte Knoten und deren Unterbäume tauschen.
Hinweis: Diese Aktion funktioniert für alle Knoten außer für die Wurzel.

- **Löschen:** Den markierten Knoten und den gesamten Unterbaum löschen.
Hinweis: Diese Aktion funktioniert bei allen Knoten inklusive der Wurzel, in dem Fall bekommt man einen leeren Baum angezeigt.
- **Knoten umbenennen und Gewicht ändern:** Nachdem ein Knoten markiert wurde, können hier die Eigenschaften des Knoten geändert werden, indem die Felder für den Knoten-Wert und den Namen ausgefüllt werden und auf "Ändern" geklickt wird.
- **Zufälliger Baum Optionen:** Hier können die Parameter für die Erstellung eines zufälligen Baumes geändert werden.
- **Minimale Tiefe:** Garantiert einen Baum von mindestens dieser Tiefe, unabhängig von der Knoten-Wahrscheinlichkeit. Es werden ganze Zahlen von 0 bis 10 akzeptiert.
- **Maximale Tiefe:** Garantiert einen Baum, der nicht tiefer als dieser Wert ist, unabhängig von der Knoten-Wahrscheinlichkeit. Es werden ganze Zahlen von 0 bis 10 akzeptiert.
- **Maximale Verzweigung:** Bestimmt die maximale Anzahl an Kindknoten pro Knoten. Es werden ganze Zahlen von 0 bis 10 akzeptiert.
- **Zufall:** Hier kann die Wahrscheinlichkeit, dass ein Kindknoten existiert, durch eine Dezimalzahl zwischen 0 und 1 angegeben werden. Durch die Eingabe 0.4 werden z.B. im Durchschnitt 40% der durch den maximalen Verzweigungsgrad beschränkten Knoten erstellt.

Hinweis: Der zufällige Baum wird mit Tiefensuche-Reihenfolge erstellt. Nach 500 erstellten Knoten werden aus Performance-Gründen keine weiteren Knoten mehr erstellt.




e. Anzeige:




 Anzeige




Text:




- ☒ Name anzeigen
- ☒ Gewicht anzeigen
- ☒ Alpha-Beta anzeigen
- ☒ Min-Max anzeigen

Skalierung:




  




  

Verschiebung:


  

Unter dem Anzeige-Reiter sind sämtliche Aktivitäten zu finden, mit denen der Nutzer das Aussehen der Webseite verändern kann.


Layout:


Layout
Leaf Even (Default) ▼


Seite:


 Akzentfarbe

Algorithmus:

 Derzeitiger Knoten

 Besuchte Knoten

 Unbesuchte Knoten

 Cutoff

Layout:

Hier sind 3 Layout-Varianten vorhanden:

- Leaf Even: Wird Standardmäßig ausgewählt. Diese Variante stellt sicher, dass die Blätter sich nicht überlappen.
- Layer Even: Die einzelnen Knoten werden auf einer Ebene komplett gleichmäßig verteilt. Dadurch kann es passieren, dass sich Kindknoten nicht mehr unterhalb, sondern weit links oder rechts vom Parent-Knoten befinden.

- Interval Splitting: Hierbei wird jedem Kindknoten rekursiv gleichmäßig viel Platz gegeben. Dadurch nutzt der Baum häufig nicht den kompletten Raum aus, da pro Ebene exponentiell weniger Platz zur Verfügung steht.

Seite:

Hier kann die Akzentfarbe der Webseite nach persönlichem Geschmack verändert werden. Durch Klicken auf den Kasten links von der "Akzentfarbe" lässt sich die Farbe von verschiedenen Komponenten der Webseite verändern.

Algorithmus:

Hier kann die Farbe des Algorithmus nach persönlichem Geschmack ausgewählt werden. Hinweis: Diese Aktion ist nur dann zu sehen, wenn eine Simulation läuft.

- Derzeitiger Knoten: Farbe des aktuell betrachteten Knoten in einer Simulation ändern.
- Besuchte Knoten: Farbe der besuchten Knoten in einer Simulation ändern. Unbesuchte
- Knoten: Farbe der unbesuchten Knoten in einer Simulation ändern. Cutoff: Farbe der
- Cutoffs in einer Simulation ändern.

Skalierung:

Unter Skalierung finden sich folgende Aktivitäten:

- Alle Knoten vergrößern bzw. verkleinern: Dies geschieht mit Hilfe vom Slider. Beim Verschieben des Kreises nach links werden Knoten kleiner, nach rechts größer.
- Baum schmaler bzw. breiter machen: Dies geschieht mit Hilfe vom Slider. Beim Verschieben des Kreises nach links werden Knoten horizontal enger stehen, nach rechts horizontal weiter entfernt stehen.
- Baum schmaller bzw. breiter machen: Dies geschieht mit Hilfe vom Slider. Beim Verschieben des Kreises nach links werden Knoten vertikal enger stehen, nach rechts vertikal weiter entfernt stehen.

Verschiebung:

Hierbei wird der Komplette Baum nach links, rechts, oben oder unten verschoben.

- Den Baum nach links bzw. rechts schieben: Dies geschieht mit Hilfe vom Slider. Beim Verschieben des Kreises nach links wird der Baum nach links verschoben, und analog nach rechts.
- Den Baum nach oben bzw. unten schieben: Dies geschieht mit Hilfe vom Slider. Beim Verschieben des Kreises nach links wird der Baum nach oben verschoben, und analog nach rechts.

Text:

Unter Text stehen vier Aktionen zur Verfügung:

- Name anzeigen: beim Klicken auf den Button werden alle Namen im Baum angezeigt bzw. versteckt.
- Gewicht anzeigen: beim Klicken auf den Button werden alle Gewichte im Baum angezeigt bzw. versteckt.

- Alpha-Beta anzeigen: beim Klicken auf den Button werden die Alpha-Beta Werte in der Simulation angezeigt bzw. versteckt.
- Min-Max anzeigen: beim Klicken auf den Button werden die Min-Max-Hilfslinien am Rand des Graphen angezeigt bzw. versteckt.

Funktionsweise der Algorithmen

Allgemein

Spielbäume

Ein Spielbaum stellt alle möglichen Entwicklungen eines Nullsummenspiels mit zwei Spielern dar. Ein Knoten im Baum repräsentiert also einen Zustand im Spiel und dessen Gewicht ist eine Bewertung der jeweiligen Stellung.

Max

Spieler 1: Möchte möglichst große Werte erzielen

Min

Spieler 2: Möchte möglichst kleine Werte erzielen

Algorithmen

Die Blattknoten bestimmen den Ausgang des Spiels, sie repräsentieren z.B. ein Matt oder Patt im Schach. Daher sollten ihre Werte immer gegeben sein. Die Algorithmen liefern nun Stellungsbewertungen für Nicht-Blattknoten und insbesondere für den Wurzelknoten. Der Wert im Wurzelknoten beschreibt schließlich den Ausgang des Spiels bei perfektem Spiel beider Seiten ausgehend von der Stellung des Wurzelknotens.

Minimax

Ausgehend von einem Knoten X wählt der Max-Spieler den Kindknoten Y mit größtem Wert aus und setzt den Wert von X auf den Wert von Y. Der Min-Spieler verfährt analog. Sollten die Kindknoten keine Blätter sein, muss der Algorithmus zunächst rekursiv auf den Kindknoten ausgeführt werden, um ihren Wert zu ermitteln.

(Jeder Zweig des Spielbaums wird von unten nach oben durchlaufen (das Spiel wird rückwärts simuliert). Wir fragen uns also, welchen Zug der Spieler zuvor gemacht hätte. Welcher Spieler am Zug ist, lässt sich an der Baumtiefe ablesen. Es werden die Gewichte der Blattknoten verglichen, um das Maximum oder Minimum zu finden (je nach Spieler). Dann wird der beste Wert an den darüberliegenden Knoten weitergegeben. Das wird wiederholt, bis der ganze Baum bestimmt ist.)

Alpha-Beta

Alpha:

Schlechtester Spielausgang für Max im betrachteten Zug (Initial: $-\infty$)

Beta:

Schlechtester Spielausgang für Min im betrachteten Zug (Initial: + unendlich)

Suchtiefe:

Bis zum wievielten Zug der Spielbaum bestimmt wird (Initial: gesamter Baum)

Je tiefer, desto genauer die Schätzung des Spielausgangs und aufwendiger die Berechnung

Ablauf:

Beim Alpha-Beta Algorithmus werden die Gewichte der Knoten mit einem Alpha oder Beta Wert verglichen (je nach Spieler). Ist das Knotengewicht besser, wird der Alpha oder Beta Wert (des Elternknotens) durch diesen ersetzt.

Zusätzlich finden Cutoffs statt. Beispiel für Beta-Cutoff: Der Alpha Wert hat sich verbessert und ist nun größer als Beta -> Max weiß, dass Min diesen Zug nicht machen würde -> restliche Kindknoten müssen nicht betrachtet werden und werden abgeschnitten

Iterative Tiefensuche

In der Iterativen Tiefensuche wird der Alpha-Beta-Algorithmus hintereinander mit wachsender Suchtiefe ausgeführt. Dafür ist es notwendig, dass auch Nicht-Blattknoten einen Wert haben. Das kann z.B. mit einer Heuristik geschehen. Nach der Ausführung des Alpha- Beta-Algorithmus für eine bestimmte Tiefe d wird der Baum so sortiert, dass die bisher am attraktivsten scheinenden Knoten als erste untersucht werden. Dadurch steigt im nächsten Alpha-Beta-Durchlauf mit Tiefe $d+1$ die Wahrscheinlichkeit eines Cutoffs und es müssen weniger Knoten betrachtet werden. Außerdem kann der Algorithmus im Gegensatz zu Minimax oder Alpha-Beta jederzeit angehalten werden und liefert trotzdem ein Ergebnis.

(Die iterative Tiefensuche funktioniert mit mehreren Iterationen des Alpha-Beta Algorithmus. Ziel ist es die Anzahl der Cutoffs zu maximieren, um weniger Zweige durchlaufen zu müssen. Dafür wird zunächst der Alpha-Beta Algorithmus mit geringer Suchtiefe durchgeführt.

Anschließend rotiert man den Baum (vertauscht die Reihenfolge einzelner Zweige), um in der nächsten Iteration (mit größerer Suchtiefe), mehr Cutoffs zu erzeugen.)