

MATTHIAS TSCHÖPE,
KUNAL OBEROI

Übung 6

Agenda



Informierte Suche



Alpha-Beta / Prunning

Informierte Suche

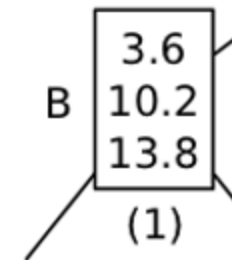
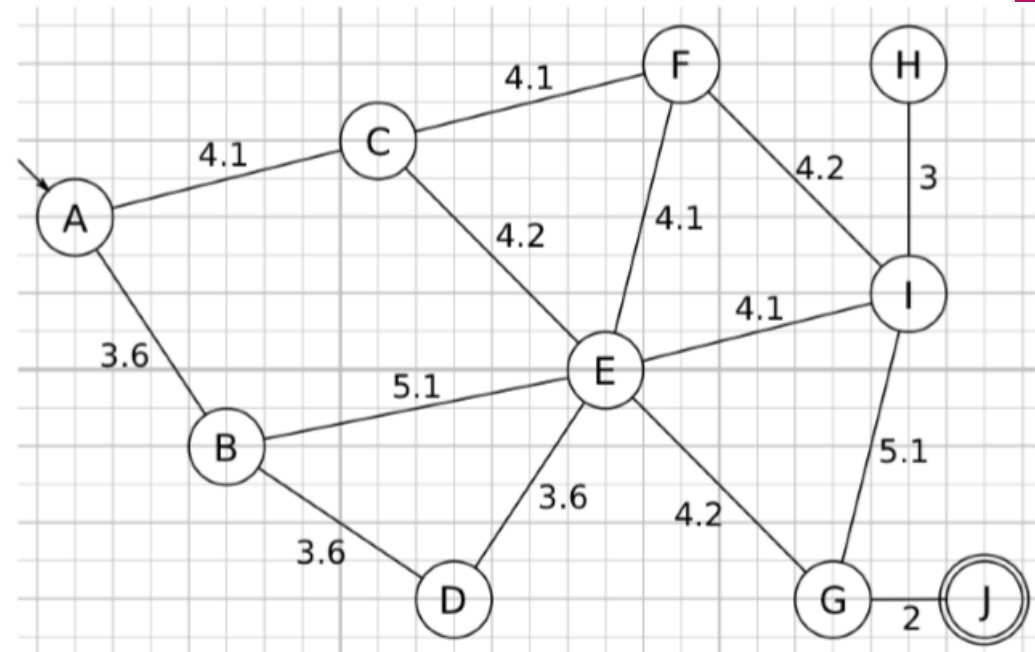
- ▶ Idee: Erweiterung des Shortest-Path (Dijkstra-Algorithmus) um einen Schätzwert (Heuristik), d.h wir können schätzen wie weit unser Zielknoten entfernt ist.
- ▶ Kostenfunktion zur Berechnung: $\text{cost}(k) = g(k) + h(k)$
- ▶ Wobei $\text{cost}(k)$ = "estimated total path value" darstellt , $g(k)$ = "cost to reach the Node", $h(k)$ = "heuristic value"(Luftlinien Distanz zwischen zwei Knoten)
- ▶ Der Algorithmus ist optimal, wenn seine Heuristic monoton ist (siehe Skript)
- ▶ Der Algorithmus ist vollständig, wenn er eine Lösung gefunden hat

Aufgabe1 (A* Algorithmus)

Verwenden Sie A* Algorithmus, um die kosten günstigste Distanz zwischen A und J zu suchen.

-Erstellen Sie ein Baum in der Notation rechts, wobei B für den Knoten steht, (1) für die Sequenznummer, 3.6 Kosten zum Knoten, Luftlinien Distanz 10.2 und Gesamtkosten

-Rechnen sie mit der Formel von Folie 4 die kostengünstigste Distanz aus



	A	B	C	D	E	F	G	H	I
$d(\cdot, J)$	13	10.2	10	7	5.8	8.1	2	7.1	4.1

Alpha / Beta Pruning (Min-Max Algorithmus)

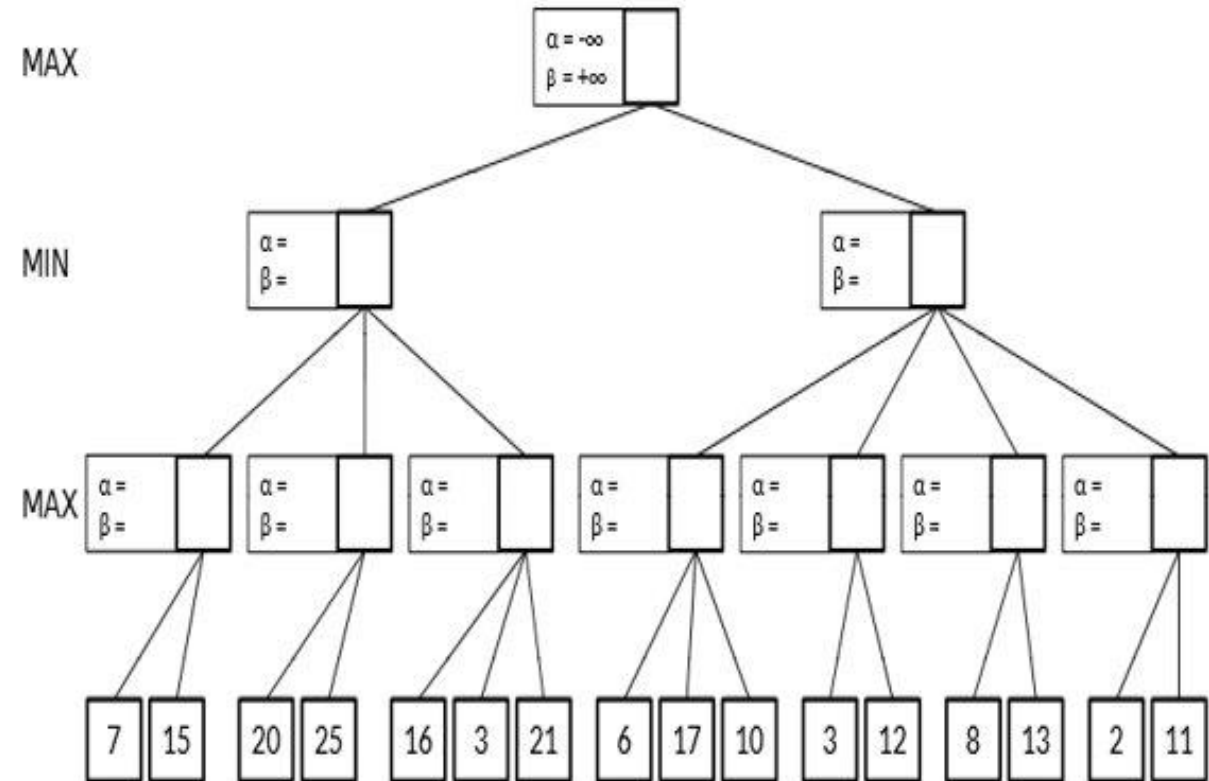
- ▶ Min-Max Such Algorithmen sind für Spiele mit zwei Spielern konzipiert z.B unser Semester Projekt (Othello)
- ▶ Spieler Min versucht, das Ergebnis des Spiels zu minimieren, Spieler Max versucht, es zu maximieren
- ▶ Max beginnt in der Regel zuerst, Alpha- und Beta Pruning kann die Anzahl der Knoten minimieren, die der Algorithmus untersucht (um das Spiel zu lösen)
- ▶ Anwendungsbeispiele : z.B in Othello oder anderen Spielen

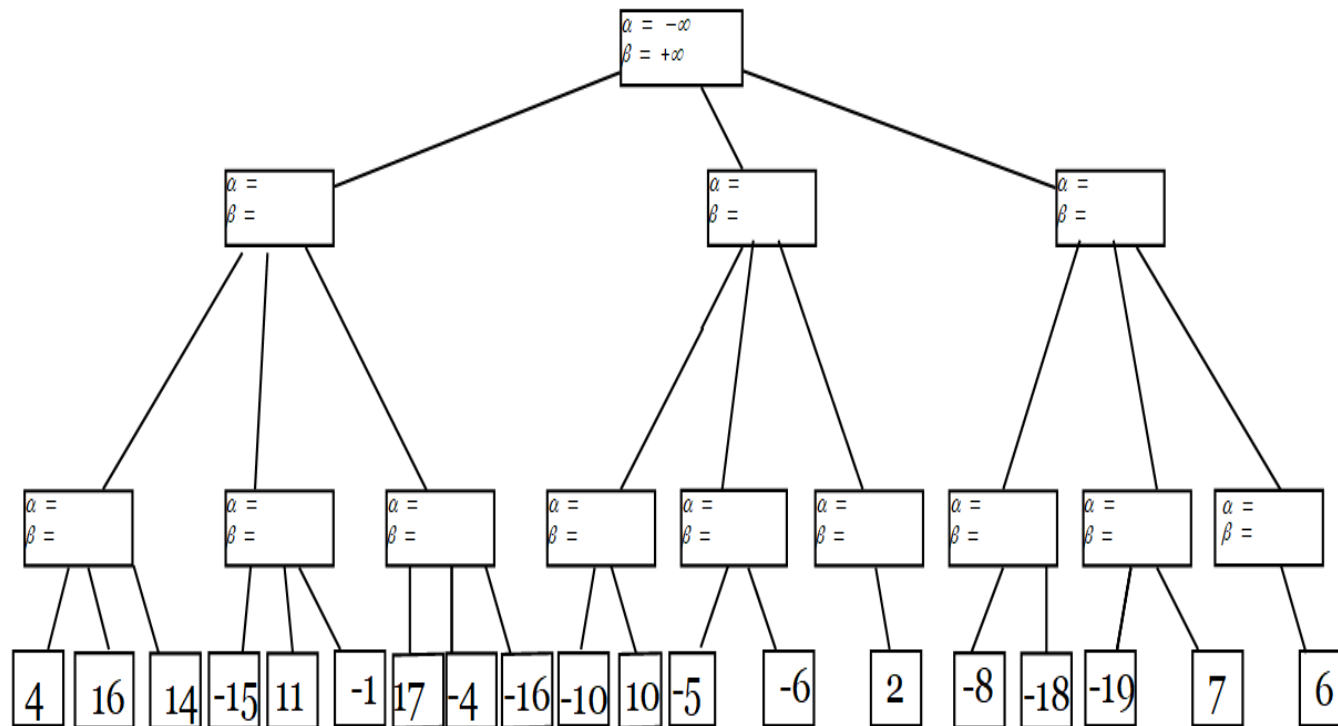
Algorithmus

- ▶ Der Algorithmus funktioniert wie folgt:
 - ▶ Der Algorithmus sucht in der Tiefe bis er einen Knoten erreicht, wo die Werte aller untergeordneten Elementen bekannt sind
 - ▶ Der Spieler, wo am Zug ist wählt den besten Wert seiner Strategie aus (maximum für MAX, minimum für MIN)
 - ▶ Auf dem Weg, berechnet Spieler Max die Werte von $\text{Beta} = \min(\text{Werte des aktuellen Knotens})$, Analog für Spieler Min $\text{Alpha} = \max(\text{Werte des aktuellen Knotens})$
 - ▶ Min Spieler sucht keine Werte aus die größer sind als Beta (analog für Max Spieler er sucht keine Werte aus die kleiner sind als Alpha)
 - ▶ Falls Max einen Knoten $\geq \text{Beta}$ findet, braucht er keine weitere Knoten mehr zu beobachten dies nennt man (Beta- Kürzung)
 - ▶ Falls Min einen Knoten $\leq \text{Alpha}$ findet, braucht er keine weitere Knoten mehr zu beobachten dies nennt man (Alpha-Kürzung)

Aufgabe 2 (Alpha / Beta Pruning)

- ▶ Füllen Sie die Werte von Alpha / Beta erst dann, wenn ein Knoten ausgewertet ist
- ▶ Füllen Sie den berechneten Wert jeden Knotens aus
- ▶ Markieren Sie Alpha / Beta durch Kürzungen (durchstreichen genügt)
- ▶ Hinweis: Initial-Wert für Alpha ist $-\infty$ und Beta $+\infty$





Aufgabe 3 (Alpha / Beta Pruning)

GLEICHES VORGEHEN WIE
IN DER AUFGABE 2 NUR
ANDERES BEISPIEL!