Grundlagen der künstlichen Intelligenz: Hausaufgabe 1

Tom Nick - 340528
Niklas Gebauer - 340942

Aufgabe 1

a) **Zustandsraum:** (a, b, c) wobei $a \in \{0, \ldots, 100\}, b \in \mathbb{N}, c \in \{A, B, C, Z, i, j\}$
wobei a die aktuelle Position, b die benötigte Zeit und c den Ladezustand beschreibt.
Anfangszustand: $(A, 0, 100)$
Zielzustand: (Z, b, c) wobei $c \geq 50$, b hat keine Einschränkung. **Aktionen:**

1. $$\mathsf{fahren}(start, ziel, zeit, energie) : (a, b, c) \rightarrow (x, y, z)$$

   mit

   $$\begin{aligned} a = start \wedge x = ziel \wedge \\ z = c - energie \wedge z \geq 0 \ \wedge y = b + zeit \wedge \\ (start, ziel, zeit, energie) \in \{&(A, Z, 170, 95), (Z, A, 170, 95), \\ &(A, i, 100, 50), (i, A, 100, 50), \\ &(i, Z, 200, 100), (Z, i, 200, 100), \\ &(i, j, 100, 50), (j, i, 100, 50), \\ &(i, B, 80, 45), (B, i, 80, 45), \\ &(j, Z, 80, 40), (Z, j, 80, 40), \\ &(j, C, 25, 20), (C, j, 25, 20), \\ &(Z, C, 20, 10), (Z, C, 20, 10)\} \end{aligned}$$

2. $$\mathsf{laden}(zustand) : (a, b, c) \rightarrow (x, y, z)$$

   mit

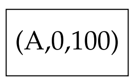
   $$zustand \in \{i, j\} \wedge y = b + 200 \wedge z = 100 \wedge zustand = a = x$$

b) **Verzweigungsgrad:** maximal 3
Tiefe: 6 wenn man sich beim Suchen intelligent anstellt. Wobei das beudeutet, dass wir einen Knoten nur 2x besuchen wenn im zweiten Besuch des Knotens die Ladung größer ist als beim ersten Besuch.

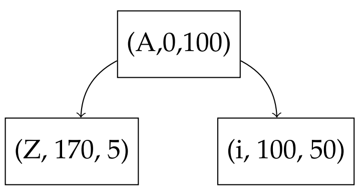
c) Da wir den schnellsten Weg Finden wollen wäre **Branch and Bound** am besten, wir haben keine Heuristiken, also kein A*, wir haben aber Pfadkosten also keine BFS/DFS.

d)

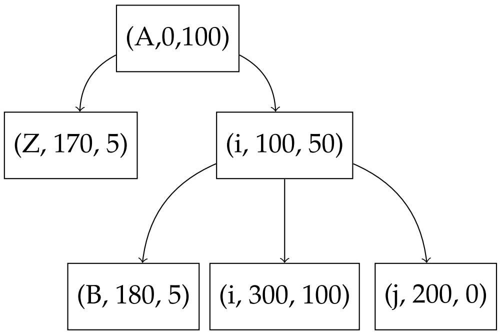
1.



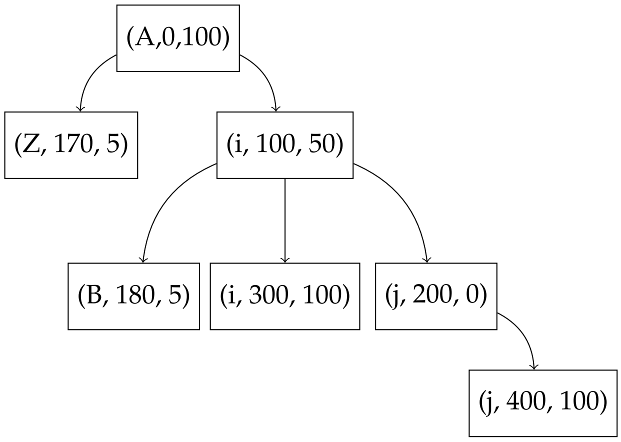
2.



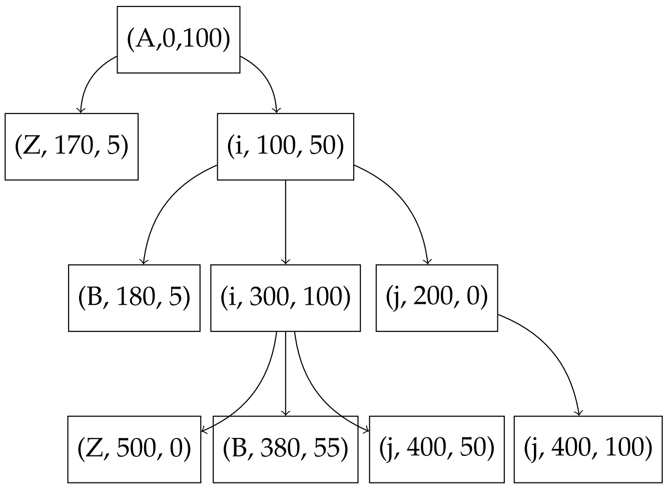
3.



4. Die dritte und vierte Expansion von dem Knoten (Z,170,5) und (B,180,5) bewirken keine Veränderung des Baumes, da Sie keine Nachfolger haben. (Energie reicht nicht aus um zu einem anderen Knoten zu fahren)
5.



6.



e) Algorithmen der dynamischen Programmierung werden genutzt um optimale Lösungen zu finden, in dem Beispiel hier wollen wir den kleinstmöglichsten Weg von A nach Z finden, wobei die verbleibende Energie größer gleich 50 ist. Dafür werden ALLE möglichen Wege untersucht, viele werden jedoch von vornerein ausgeschlossen (eine Art intelligentes Brute Force). In diesem Beispiel benutzen wir Branch and Bound. Dieser Algorithmus untersucht als nächsten Knoten immer den mit den niedrigsten insgesamten Pfadkosten, dadurch werden gleichzeitig viele unnütze Wege, Schleifen... ausgeschlossen.

Aufgabe 2

a) **Zustandsraum:** $((w_1, x_1, y_1), (w_2, x_2, y_2), (w_3, x_3, y_3), (w_4, x_4, y_4), (w_5, x_5, y_5))$ wobei $x_i, y_i \in \{(a, b) \mid a \in \{1, \ldots, 5\}, b \in \{1, \ldots, 5\}\}$ mit $i \in \{1, \ldots, 5\}$
Wir können maximal 5 Stapel bilden, da wir 5 Kisten haben. Jede Kiste wird beschrieben durch ein Tupel

von x und y welche den Stapel sowie die Höhe in diesem Stapel beschreiben.w_1 beschreibt den Wert der Kiste, man könnte auch die Reihenfolge dafür benutzen, aber dadurch werden die Formalitäten kniffliger.
Anfangszustand: $((1,1,2),(2,1,1),(3,1,4),(4,1,3),(5,1,5))$
Zielzustand: $((1,x_1,5),(2,x_2,4),(3,x_3,3),(4,x_4,2),(5,x_5,1))$ mit $x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 = x_4 \wedge x_4 = x_5$
Aktionen:

1.

$$\mathsf{bewegen}() : (von, nach, (w_1,x_1,y_1),(w_2,x_2,y_2),(w_3,x_3,y_3),(w_4,x_4,y_4),(w_5,x_5,y_5)) \to (c_1,a_1,b_1),(c_2,a_2,b_2),(c_3,a_3,b_3$$

mit

1. Es gibt x_i mit $i \in \{1,\dots,5\}$ sodass $von = i$.
2. Sei $\phi = (von, x, y_i)$ mit $i \in \{1,\dots,5\}$ und i maximal. D.h. es darf kein (von, x, y_i) existieren mit $y_i > \phi_2$. Sei $\lambda = (nach, x, y_i)$ mit $i \in \{1,\dots,5\}$ und i maximal. D.h. es darf kein $(nach, x, y_i)$ existieren mit $y_i > \lambda_2$. Es muss gelten:
$$(a_{\phi_2}, a_{\phi_2}, a_{\phi_2}) = (\phi_1, nach, \lambda_3 + 1)$$
Alle anderen Werte ändern sich nicht.

Aufgabe 3

Aufgabe 4