

Beschreibung

In diesem Projekt nehmen Sie die Rolle einer Hafenverwaltungsfirma Colorbox ein. Ihr Ziel ist es einen Kran zu programmieren, welcher die Container nach Farbe sortiert, damit die Container effizient auf Schiffe verladen werden können. Jede Farbe entspricht hier einer Destination, zu welcher ein Container verfrachtet werden sollte.



Als Notation für dieses Problem verwenden wir die Position der Stellplätze. In der Vorlage haben Sie drei Stellplätze. Wir teilen jedem Stellplatz von links aus eine Zahl beginnend bei 0 zu. Die Stellplätze sind also 0, 1, 2. Wenn Sie einen Container von Stellplatz 1 zu Stellplatz 0 bewegen, wäre die Notation dazu $(1, 0)$ (von Platz 1 zu Platz 0). Ein Container kann nicht von einem Stellplatz zum gleichen Stellplatz gelegt werden, da dann ja der gleiche Zustand erreicht würde. Das bedeutet, das Beispielsweise der Zug $(0, 0)$ nicht relevant ist. Im gezeigten Beispiel gäbe es vier legale Züge: $(0, 1)$, $(0, 2)$, $(1, 1)$, $(1, 2)$. Die Züge $(2, 0)$ und $(2, 1)$ wären nicht legal, da auf dem Stellplatz 2 kein Container steht.

Sie beginnen bereits mit einer implementierten Suche in der Codevorlage. Diese Suche ist jedoch extrem ineffizient. Nach 5000 durchsuchten Zuständen bricht das Programm ab, um Ihren Computer zu schonen. Es kann also sein, dass die Suche keine Lösung findet.

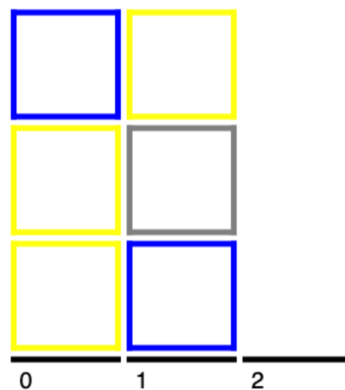


Abbildung 1: Screenshot von WebTigerJython. Die Stellplätze sind unten (schwarz) und die Container stehen auf den Stellplätzen 0 und 1. Der Stellplatz 2 ist hier leer.

Regeln

Ziel: Auf jedem Stellplatz sollen immer nur Container einer Farbe stehen, damit Sie effizient verladen werden können.

Züge: Ihr Kran kann jeweils nur einen einzigen Container aufheben und an eine andere Position bewegen.

Aufgabe

Hier werden zusätzliche Aufgaben beschrieben, welche spezifisch für Ihr Projekt sind. Die Aufgaben sind in zwei Gruppen unterteilt: Pflichtaufgaben und optionale Aufgaben. Die Pflichtaufgaben sollen am Ende des Projektes beantwortet sein. Die optionalen Aufgaben dienen dazu, sich weiter im Thema zu vertiefen. Diese Aufgaben sind Vorschläge. Sie dürfen auch eigene Fragestellungen vertiefen. Deklarieren Sie Ihre Fragestellungen klar im Projektbericht. Für alle Projekt gilt natürlich, dass Sie zuerst die allgemeine Aufgabenstellung und den Code (falls vorhanden) verstehen sollten, bevor Sie die untenstehenden Aufgaben bearbeiten.

Pflichtaufgaben

- In der Vorlage ist bereits eine Suche implementiert. Vergleichen Sie diese Suche mit dem manuellen Eingeben einer Lösung. Was fällt auf? Um was für einen Algorithmus könnte es sich handeln?
- Schreiben Sie ein Programm (oder verbessern Sie das existierende), welches jeweils eine optimale Lösung für Ihr Problem findet.
- (Falls Sie eine Breitensuche implementiert haben): Erweitern Sie das Problem, indem Sie in der Konfiguration mehr Container hinzufügen. Finden Sie mit der Breitensuche noch eine Lösung? Begründen Sie.

Optionale Aufgaben

- Implementieren Sie eine Heuristik, welcher jedem Zug angibt, wie gut er ist. Begründen Sie, warum Sie Ihre Heuristik gewählt haben.
- Implementieren Sie Ihre Heuristik als in einem Suchalgorithmus.
- Was passiert mit dem Problem, Wenn Ihr Kran nun mehr als einen Container von oben gleichzeitig aufheben kann?
- (Sehr schwierig!) Implementieren Sie eine Suche, welche das Aufheben mehrerer Container in betracht zieht.