

Coding Challenge (tipps && tricks)

Ein paar nützliche Hinweise

Ausgaben mit hoher Geschwindigkeit

Debug-Ausgaben

- `System.err.println()` ist langsam
 - Quatschen mit dem Betriebssystem
- Kartenausgabe: Hunderte Aufrufe
 - Ausgabe großer Karten ist langsam
 - Timeout?

StringBuilder

- StringBuilder ist eine Standard-Javaklasse
 - `StringBuilder sb = new StringBuilder();`
 - `sb.append("Hallo Welt");`
 - Tausendmal Zeugs dranhängen
 - `"\n"` um neue Zeile einzufügen
 - `System.err.println(sb.toString());`
 - Nur **einmal** eine Ausgabe machen, viel schneller

Das Pac-Man-Syndrom

Spielfeld ist „unendlich“

- Das Spielfeld muss keine Außenwand haben
 - Wer am Rand rausläuft, der landet am gegenüberliegenden Rand
 - Bot sieht auch Karteninformation über Rand hinweg
 - Illusion einer Welt, die sich unendlich oft wiederholt
- Ihre Karte ist aber nicht unendlich groß, wie korrekt adressieren?

Koordinaten berichtigen (1/2)

- Angenommen, x sei 0, Breite sei 5
 - Bot steht ganz links
 - Dann nochmal nach links \rightarrow x wird -1
 - -1 ist aber kein gültiger Array-Index
 - `ArrayIndexOutOfBoundsException`, igit
 - Lösung: Breite der Karte aufaddieren
 - Angenommen Breite sei 5, dann nun $x = 4$ (ganz rechts)

Koordinaten berichtigen (2/2)

- Angenommen, x sei 4, Breite sei 5
 - Bot steht ganz rechts
 - Dann nochmal nach rechts $\rightarrow x$ wird 5
 - 5 ist aber kein gültiger Array-Index
 - `ArrayIndexOutOfBoundsException`, igit
 - Lösung: Koordinate Modulo Breite nehmen
 - x sei 5, x modulo Breite $\rightarrow x$ wird 0 (ganz links)

Kombinierte Lösung

- $x = ((x + richtung) + breite) \% breite;$
 - $richtung = -1 \rightarrow$ nach links
 - $richtung = 1 \rightarrow$ nach rechts
- Für y entsprechend mit der Höhe
- Entsprechende Hilfsmethode schreiben, um die Rechnung nicht an dutzenden Stellen zu haben

Kartennavigation

Zielstellung

- Es sollen kürzeste Wege zu Zielen auf einer Karte gefunden werden.
- Kürzeste Wege zu beliebigen Zielen auf der Karte lassen sich ganz einfach planen, wenn die Entfernung zum Bot für jedes Feld bekannt ist.

Idee (1/2)

- 2D-Karte von Entfernungen
- Erstmal ist jedes Feld „unendlich weit weg“
- Wenn ein nicht-Wand-Feld einen Nachbarn mit geringerem Abstand zum Bot hat:
 - Abstand des Felds ist $\text{<Nachbarabstand>} + 1$
 - Das Nachbarfeld ist ja einen Schritt entfernt.

Idee (2/2)

- Wenn die Karte ein 2D-Array ist, dann läuft man einfach so oft durch das Array und aktualisiert die Abstände, bis keine Verbesserungen mehr eintreten.

Kartenbeispiel

#	#	#	#	#
#				#
#		#	b	#
#				#
#	#	#	#	#

Wand (oder unbekannt)
b Bot

Entfernungskarte

∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞

Am Anfang ist alles „unendlich“ weit weg (oder eine „sehr große Zahl“)

Entfernungskarte

∞	∞	∞	∞	∞
∞	∞	∞	∞	∞
∞	∞	∞	0	∞
∞	∞	∞	∞	∞
∞	∞	∞	∞	∞

Das Feld, wo der Bot steht, ist 0 Schritte entfernt

Entfernungskarte

Beispielhaft: Von links
nach rechts, von oben
nach unten gehen.

∞	∞	∞	∞	∞
∞	∞	∞	1	∞
∞	∞	∞	0	∞
∞	∞	∞	1	∞
∞	∞	∞	∞	∞

Jetzt durch das Array gehen.

Wenn keine Wand, dann:

Niedrigsten Nachbarwert
heraussuchen.

Nachbarwert + 1 rechnen.

Wenn das besser ist als
aktuelle Bewertung, dann:

Nachbarwert + 1 übernehmen

Entfernungskarte

Beispielhaft: Von links
nach rechts, von oben
nach unten gehen.

∞	∞	∞	∞	∞
∞	∞	2	1	∞
∞	∞	∞	0	∞
∞	∞	2	1	∞
∞	∞	∞	∞	∞

Jetzt durch das Array gehen.

Wenn keine Wand, dann:

Niedrigsten Nachbarwert
heraussuchen.

Nachbarwert + 1 rechnen.

Wenn das besser ist als
aktuelle Bewertung, dann:

Nachbarwert + 1 übernehmen

Entfernungskarte

Beispielhaft: Von links
nach rechts, von oben
nach unten gehen.

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Jetzt durch das Array gehen.

Wenn keine Wand, dann:

Niedrigsten Nachbarwert
heraussuchen.

Nachbarwert + 1 rechnen.

Wenn das besser ist als
aktuelle Bewertung, dann:

Nachbarwert + 1 übernehmen

Entfernungskarte

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Wenn sich bei einem Durchlauf keine weitere Verbesserung für irgendein Feld ergibt: Fertig!

Nun ist der Abstand zu jedem Feld auf der Karte bekannt!

Entfernungskarte

Bot möchte zu diesem Feld. Von dort aus den niedrigsten Zahlen folgen!

(Bei Gleichstand:Egal!)

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Um ein bestimmtes Feld anzusteuern: Auf der Entfernungskarte beim Ziel starten, dann immer den niedrigsten Zahlen entlang!

(Rückwärts vom Ziel zum Bot laufen)

Entfernungskarte

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Um ein bestimmtes Feld anzusteuern: Auf der Entfernungskarte beim Ziel starten, dann immer den niedrigsten Zahlen entlang!

(Rückwärts vom Ziel zum Bot laufen)

Entfernungskarte

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Um ein bestimmtes Feld anzusteuern: Auf der Entfernungskarte beim Ziel starten, dann immer den niedrigsten Zahlen entlang!

(Rückwärts vom Ziel zum Bot laufen)

Entfernungskarte

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Um ein bestimmtes Feld anzusteuern: Auf der Entfernungskarte beim Ziel starten, dann immer den niedrigsten Zahlen entlang!

(Rückwärts vom Ziel zum Bot laufen)

Entfernungskarte

∞	∞	∞	∞	∞
∞	3	2	1	∞
∞	4	∞	0	∞
∞	3	2	1	∞
∞	∞	∞	∞	∞

Abstand Null, wir sind nun an der Position des Bots!

Der **letzte Schritt zum Bot** war Richtung Süden, der Bot sollte also nach Norden gehen!

(Wir sind ja vom Ziel aus rückwärts zum Bot gelaufen)

Wann die Entfernungskarte neu berechnen?

- Immer wenn sich die Karte geändert (man könnte ja eine neue Passage gefunden haben) oder der Bot eine neue Position hat.
 - Also im Endeffekt in jeder Runde, nach Update von Bot-Position und Karte.

Bester Algorithmus?

- Der Dijkstra-Algorithmus ist noch einen Tick schneller, weil nicht wiederholt durch ein Array durchgegurkt wird.
 - Wenn Sie also lieber den umsetzen wollen:
Natürlich dürfen Sie!

Karte gezielt explorieren

Karte gezielt aufdecken

- Wenn kein „besseres“ Ziel (z.B. bestimmtes Formular) bekannt: Laufen Sie immer zum nächsten bekannten nicht-Wand-Feld mit mindestens einem unbekannten Nachbarfeld.
 - Damit decken Sie garantiert die ganze Karte auf
 - Irgendwann gibt es keine bekannten Felder mit unbekannten Nachbarn mehr → Exploration fertig!