

Aufgabe 3: Zauberschule

Team-ID: 01004

Team-Name: Was?

Bearbeiter dieser Aufgabe:

Mathieu de Borman

19. November 2024

Inhaltsverzeichnis

1 Lösungsidee-----	1
2 Umsetzung-----	1
3 Beispiele-----	2
4 Quellcode-----	4

1 Lösungsidee

Das Programm markiert jedes Feld mit der Zeit die es braucht um dorthin zu gelangen. Anschließend wird von B aus der kürzeste Pfad zurückverfolgt.

Teilaufgabe 1

Das Programm zählt immer die Zeit um eine Sekunde hoch. Jedes Mal wenn eine Sekunde verstreicht, wird bei jedem Feld, welche letzte Sekunde eine Zeit bekommen hat die umliegenden leeren Felder mit der aktuellen Zeit markiert. Bei jedem Feld, das vor drei Sekunden eine Zeit bekommen hat wird dem gegenüberliegenden Feld, wenn es leer ist, die aktuelle Zeit gegeben. Das wird solange gemacht, bis das Zielfeld B eine Zeit bekommen hat.

Teilaufgabe 2

Anschließend setzen wir den Pointer auf die Stelle B. Und den Zeitzähler auf die Zeit die B bekommen hat. Nun können wir immer um die fünf stellen um den Pointer schauen. Wenn einer der fünf Felder um eins bzw. drei kleiner ist, als der Zeitzähler, dann wird diese Stelle als erstes mit einem Pfeil in die eigene Richtung oder einem Ausrufezeichen ersetzt. Im Anschluss wird der Pointer auf die neue Stelle gesetzt und der Zeitzähler auf die Zeit der neuen Stelle. Das wird solange gemacht, bis der Zeitzähler auf 0 ist bzw. der Pfad bis zum Start A zurückverfolgt wurde.

2 Umsetzung

Die Lösungsidee wird in JavaScript implementiert. Das Programm verwendet zentral zwei While Schleifen: Die erste Schleife um allen Feldern eine Zeit zu geben und die Zweite um den Pfad zurückzufinden.

In der ersten While Schleife verwenden wir bei jeder Iteration ineinander geschachtelte for Schleifen um jedes Feld überprüfen zu können. Im Anschluss wird bei jedem Feld überprüft, ob es im ersten Stockwerk die Zeit minus eins hat. Wenn das der Fall ist, dann wird bei diesem Feld überprüft, ob die Umliegenden Felder leer sind. Wenn das der Fall ist, dann wird dem benachbarten Feld die aktuelle Zeit gegeben. Gleiches Passiert für das zweite Stockwerk. Im Anschluss wird das gleiche für das zweite Stockwerk gemacht. Zudem wird bei jedem Feld auch noch überprüft, ob es drei Sekunden alt ist. Wenn das der Fall ist, dann wird bei dem gleichen Feld auf dem anderen Stockwerk überprüft, ob es leer ist. Wenn das der Fall ist, dann bekommt das Feld auf dem anderen Stockwerk die aktuelle Zeit. Die While Schleife wird unterbrochen sobald das Feld B eine Zeit bekommen hat.

In der zweiten While Schleife Wird bei jeder Iteration zuerst überprüft, ob sich die aktuelle Position in dem ersten Stockwerk befindet, wenn das der Fall ist, dann werden für die Felder über, unter, rechts und links neben dem aktuellen Feld überprüft, ob die Zeit dort um eins kleiner ist, als die des aktuellen Feldes. Wenn das der Fall ist, dann wird dieses Feld mit einem Pfeil in die eigene Richtung markiert und zur neuen Position gemacht. Danach wird überprüft, ob die gleiche stelle im anderen Stockwerk die Sekunden älter ist, als die aktuelle. Wenn das der Fall ist, dann Wird auch hier ein Pfeil (also ein Ausrufezeichen) in die eigene Richtung gemacht und das Feld wird zur neuen Position. Das gleiche wird mit dem zweiten Stockwerk gemacht. Die While Schleife endet sobald der Pfad bis A zurückverfolgt wurde bzw. die Anzahl der Sekunden um auf das aktuelle Feld zu komme 0 ist.

3 Beispiele

Wir rufen nun das JavaScript-Programm mit den verschiedenen BWINF-Eingabedateien auf. Diese Dateien sind in demselben Ordner wie die Programmdatei. Das Programm wird mit Hilfe des Browsers ausgeführt. In dem Dialogfeld kann man dann den Plan der Zauberschule eingeben.

Lösungen:

Zauberschule0.txt

. . . . # #
. ##### . # . ##### . #
. . . # . # . . . # . #
. # . ##### . # . #
. . . # # . #
. ##### ##### ##### . #
. #

. # . #
. . . # . # . # . . . #
. . . # . # . # . . .

. # # # . # . # . # # #
. . . . # . # . . . #
. # # # . . . #
. . . . # #
. # # # # # # # # . #
. . . # > > ! #
. # . # . # . # # # . #
. # . . . # . . . # . #
#

Man braucht 8 sekunden um von A nach B zu kommen.

Zauberschule1.txt

. . . # # . . . # #
. # . # . # ## . # . # . # . # ## . #
. # . # . . . # . # . # . . . # . . . #
. # ## . # . # . # ## ## ## . # ## #
. # . # . . . # . # B # . . . #
. # . # . # ## . # ^ # ## . # ## ## ##
. # . . . # . # . # ^ < < # #
. # ## ## . # . # ## # ## ## ## ## . #
. #

. # # #
. # ## . # . # . # ## . # . # ## . # . #
. # . # # #
. # . # ## # ## ## ## . # . #
. # . # # #
. # ## . # . # . # ## . # ## . # . # . #
. # . # . . . # . # . . . # #
. # . # ## ## ## . # ## . # ## . # ## . #
.

Man braucht 4 sekunden um von A nach B zu kommen.

Zauberschule2.txt

Zauberschule3.txt


```
# . . . # . # . . . . # . # . . . . # . # . # . # . #  
# . # . # # # # # # . # . # . # # # # # . # . # . # . #  
# . # . . . . . # . # . . . # . # . . . # . # . . . #  
# . # # # . # # # # # . # # # # # . # # # # # . # # # # #  
# . . . # . . . . . . . . # . . . . . . . . #  
# # # # # # # # # # # # # # # # # # # # # # # # # #  
Man braucht 28 sekunden um von A nach B zu kommen.
```

Zauberschule4.txt

Externe Datei:

Zauberschule4_Lösung.txt

Man braucht 84 sekunden um von A nach B zu kommen.

Zauberschule5.txt

Externe Datei:

Zauberschule5_Lösung.txt

Man braucht 124 sekunden um von A nach B zu kommen.

4 Quellcode

```
let bild = document.getElementById("bild")  
  
let context = bild.getContext("2d")  
  
let eingabe = prompt("Zauberschule") //Dialogfeld für die eingabe  
  
let array2 = []  
  
let teil = []  
  
let bpos = []  
  
let zweites_Stockwerk = []  
  
array = eingabe.split(" ") //Eingabe wird in einzelne Felder geteilt  
  
let Höhe = array.shift() //die Höhe wird sich gemerkt  
  
let Breite = array.shift() //die Breite wird sich gemerkt  
  
for (let x = 0; x < Höhe * 2 + 1; x++) { //Felder werden in einer zweidimensionalen Liste gespeichert  
  
    array2 = []  
  
    teil = array[x].split("")  
  
    for (let y = 0; y < Breite; y++) {
```

```

if (array[x][y] == " " || array[x][y] == "") {
} else if (array[x][y] == "A") { //der Stelle A wird der Wert 0 gegeben
    array2.push(0)
} else if (array[x][y] == "B") { //die Stelle B wird sich gemerkt und zu einem leeren Feld gemacht
    bpos.push(x)
    bpos.push(y)
    array2.push(".")
} else {
    array2.push(array[x][y])
}
}

zweites_Stockwerk.push(array2)
}

let erstes_Stockwerk = []
for (let zähler = 0; zähler < höhe; zähler++) { // die Zauberschule wird in zwei Stockwerke aufgeteilt
    erstes_Stockwerk.push(zweites_Stockwerk.shift())
}
zweites_Stockwerk.shift()
let zähler = 0
while (erstes_Stockwerk[bpos[0]][bpos[1]] == ".") {
    //die Felder werden mit der Zeit die es braucht um zu dem Feld zu kommen markiert
    zähler++
    for (let x = 0; x < höhe; x++) {
        for (let y = 0; y < breite; y++) {
            if (zähler == erstes_Stockwerk[x][y] + 1) { //erstes Stockwerk
                if (erstes_Stockwerk[x + 1][y] == ".") { //Über dem eigenen Feld
                    erstes_Stockwerk[x + 1][y] = zähler
                }
                if (erstes_Stockwerk[x - 1][y] == ".") { //Unter dem eigenen Feld
                    erstes_Stockwerk[x - 1][y] = zähler
                }
            }
        }
    }
}

```

```

if (erstes_Stockwerk[x][y + 1] == ".") { //Rechts neben dem eigenen Feld
    erstes_Stockwerk[x][y + 1] = zähler
}

if (erstes_Stockwerk[x][y - 1] == ".") { //Links neben dem eigenen Feld
    erstes_Stockwerk[x][y - 1] = zähler
}

}

if (zähler == zweites_Stockwerk[x][y] + 1) { //zweites Stockwerk
    if (zweites_Stockwerk[x + 1][y] == ".") { //Über dem eigenen Feld
        zweites_Stockwerk[x + 1][y] = zähler
    }

    if (zweites_Stockwerk[x - 1][y] == ".") { //unter dem eigenen Feld
        zweites_Stockwerk[x - 1][y] = zähler
    }

    if (zweites_Stockwerk[x][y + 1] == ".") { //Rechts neben dem eigenen Feld
        zweites_Stockwerk[x][y + 1] = zähler
    }

    if (zweites_Stockwerk[x][y - 1] == ".") { //Links neben dem eigenen Feld
        zweites_Stockwerk[x][y - 1] = zähler
    }
}

if (zähler == erstes_Stockwerk[x][y] + 3) { //erstes zu zweites Stockwerk Stockwerkwechsel
    if (zweites_Stockwerk[x][y] == ".") {
        zweites_Stockwerk[x][y] = zähler
    }
}

if (zähler == zweites_Stockwerk[x][y] + 3) { //zweites zu erstes Stockwerk Stockwerkwechsel
    if (erstes_Stockwerk[x][y] == ".") {
        erstes_Stockwerk[x][y] = zähler
    }
}

```

```

    }

}

}

bpos.push(true) //starte auf dem ersten Stockwerk

let wert = erstes_Stockwerk[bpos[0]][bpos[1]]

console.log("Man braucht",wert,"sekunden um von A nach B zu kommen")

//schreibe in die console wie lange es braucht um von A nach B zu kommen

erstes_Stockwerk[bpos[0]][bpos[1]] = "B"

while (wert != 0) { //den weg zurückverfolgen

    if (bpos[2]) { //erstes Stockwerk

        if (erstes_Stockwerk[bpos[0] + 1][bpos[1]] == wert - 1) { //Über dem aktuellen Feld

            erstes_Stockwerk[bpos[0] + 1][bpos[1]] = "^"

            bpos[0]++

            wert--

        }

        if (erstes_Stockwerk[bpos[0] - 1][bpos[1]] == wert - 1) { //unter dem aktuellen Feld

            erstes_Stockwerk[bpos[0] - 1][bpos[1]] = "v"

            bpos[0]--

            wert--

        }

        if (erstes_Stockwerk[bpos[0]][bpos[1] + 1] == wert - 1) { //Rechts neben dem aktuellen Feld

            erstes_Stockwerk[bpos[0]][bpos[1] + 1] = "<"

            bpos[1]++

            wert--

        }

        if (erstes_Stockwerk[bpos[0]][bpos[1] - 1] == wert - 1) { //Links neben dem aktuellen Feld

            erstes_Stockwerk[bpos[0]][bpos[1] - 1] = ">"

            bpos[1]--

            wert--

        }

    }

    if (zweites_Stockwerk[bpos[0]][bpos[1]] == wert - 3) { //zum zweiten Stockwerk wechseln

```

```

zweites_Stockwerk[bpos[0]][bpos[1]] = "!"

bpos[2] = false

wert = wert - 3

}

} else { //zweites Stockwerk

if (zweites_Stockwerk[bpos[0] + 1][bpos[1]] == wert - 1) { //Über dem eigenen Feld

zweites_Stockwerk[bpos[0] + 1][bpos[1]] = "^"

bpos[0]++

wert--

}

if (zweites_Stockwerk[bpos[0] - 1][bpos[1]] == wert - 1) { //unter dem eigenen Feld

zweites_Stockwerk[bpos[0] - 1][bpos[1]] = "v"

bpos[0]--

wert--


}

if (zweites_Stockwerk[bpos[0]][bpos[1] + 1] == wert - 1) { //rechts neben dem aktuellen Feld

zweites_Stockwerk[bpos[0]][bpos[1] + 1] = "<"

bpos[1]++


wert--


}

if (zweites_Stockwerk[bpos[0]][bpos[1] - 1] == wert - 1) { //links neben dem aktuellen Feld

zweites_Stockwerk[bpos[0]][bpos[1] - 1] = ">"

bpos[1]--


wert--


}

if (erstes_Stockwerk[bpos[0]][bpos[1]] == wert - 3) { // zum ersten Stockwerk wechseln

erstes_Stockwerk[bpos[0]][bpos[1]] = "!"

bpos[2] = true

wert = wert - 3


}

}

```

```

}

var table = document.createElement("table") //erstelle Tabelle

var thead = document.createElement("thead")

var headerRow = document.createElement("tr")

for (let x of erstes_Stockwerk) { //gib das erste Stockwerk in die Tabelle ein

    var row1 = document.createElement("tr")

    for (let y of x) {

        var cell1 = document.createElement("td")

        if (y > 0) { // alle zahlen werden durch leere Felder ersetzt

            cell1.textContent = "."

        } else {

            cell1.textContent = y

        }

        row1.appendChild(cell1)

    }

    table.appendChild(row1)

}

var row1 = document.createElement("tr")

var cell1 = document.createElement("td")

cell1.textContent = '\u00c2'

row1.appendChild(cell1)

table.appendChild(row1)

for (let x of zweites_Stockwerk) { //gib das zweite Stockwerk ein

    var row1 = document.createElement("tr")

    for (let y of x) {

        var cell1 = document.createElement("td")

        if (y > 0) { // alle zahlen werden durch leere Felder ersetzt

            cell1.textContent = "."

        } else {

            cell1.textContent = y

        }

    }

}

```

```
    row1.appendChild(cell1)
}

table.appendChild(row1)

}

document.body.appendChild(table) // füge die Tabelle dem dokument hinzu
```