

# Aufgabe 1: Arukone

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-----	3

## 1 Lösungsidee

Es werden immer wieder zufällige Wege in die Karte eingezeichnet. Der Anfang sowie das Ende des Weges werden mit der Zahl des Weges ausgestattet und der Rest wird als leeres Feld behandelt. Anschließend wird der Prozess solange wiederholt, bis aus allen unbenutzten Feldern entweder leere oder Felder mit Zahlen wurden. Danach werden Zahlen die nur einmal vorkommen aus der Karte entfernt und durch eine leere Stelle ersetzt.

## 2 Umsetzung

Die Lösungsidee wird in JavaScript implementiert. Das Programm verwendet eine Zentrale While Schleife um die Wege zu erstellen und eine Zentrale for Schleife um die zu kurzen Wege zu entfernen.

Die While Schleife wird solange ausgeführt bis alle Felder eine Zahl bekommen haben. Am Anfang der Schleife wird der Anfangspunkt des Weges auf einen unbenutzten Punkt der Karte gesetzt.

Anschließend wird innerhalb einer for Schleife der Weg solange verlängert, bis die Länge des Weges der Breite der Karte entspricht. Innerhalb dieser for Schleife wird als erstes überprüft, ob die Position sich geändert hat. Wenn das der Fall ist, wird überprüft, ob es die erste Stelle des Weges ist, ist das der Fall, wird die Stelle mit der Wegzahl markiert. Wenn es nicht die erste Stelle ist, wird es mit 0 markiert. Danach wird mit einer 50% Wahrscheinlichkeit probiert sich um ein Feld nach oben zu bewegen. Wenn es probiert wird, wird zuerst überprüft, ob es obendrüber ein Feld gibt und wenn es eines gibt, ob dieses unbenutzt ist. Wenn es das Feld nicht gibt oder schon benutzt wurde, dann wird sich gemerkt, dass man nicht nach oben gehen kann. Wenn das Feld obendrüber nun aber unbenutzt ist, dann wird dieses Feld zur neuen Position, es wird sich gemerkt, dass die Stelle sich bewegt hat. Anschließend wird das gleiche für die Richtungen Rechts, Unten und Links gemacht. Wenn keine der vier Richtungen zufällig nicht überprüft wurden oder es in die Richtung, in die es probiert wurde nicht ging, dann wird der Weg als nicht verlängert gespeichert. Wenn es in alle vier Richtungen nicht geht,

dann wird die for Schleife beendet. Danach wird innerhalb der for Schleife überprüft, ob es überhaupt noch freie Felder gibt. Wenn es keine Freien Felder mehr gibt, dann wird die While schleife unterbrochen. Wenn die for Schleife zu ende ist, wird die letzte Stelle des Weges mit der Weg zahl markiert.

Die Anschließende for Schleife wird für jeden gezeichneten Weg ausgeführt. Innerhalb dieser for Schleife wird dann durch zwei ineinander geschachtelter for Schleifen durch jedes Feld in der Karte gegangen. Wenn die Nummer des Feldes, die gleiche ist wie die des Weges nach dem gesucht wird, dann wird sich gemerkt, dass dieser Weg ein Weiteren Endpunkt hat. Außerhalb der zwei for Schleifen, die Jedes Feld durchgehen, wird überprüft, ob nur ein Endpunkt gefunden wurde. Wenn der Weg also nur eine Stellelang ist, dann wird dieser entfernt.

### 3 Beispiele

Wir rufen nun das JavaScript-Programm auf. Das Programm wird mithilfe des Browsers ausgeführt.

Die Rätsel, Lösungen, und die Lösungen der Algonautin sind in der Zip Datei Beigefügt.

Hier sind Drei etwas kleinere Rätsel bei welchen die verschiedenen Lösungen auch in den Dateien sind.

#### Rätsel\_1

4

4

1 0 3 4

0 3 2 0

0 0 0 4

0 1 0 2

#### Rätsel\_2

7

1

0

1 0 0 0 8 8 2

0 0 0 10 3 0 0

0 0 10 6 0 0 0

0 0 0 0 6 0 0

4 9 1 5 5 0 2

0 0 0 0 0 0 7

4 0 9 0 0 3 7

## Rätsel\_3

8

1

7

1 0 0 0 13 10 16 16

0 17 0 0 13 10 8 8

14 17 0 0 3 0 0 4

14 9 0 0 0 0 0 0

12 9 1 0 0 0 0 4

12 6 7 7 0 0 2 15

11 0 0 0 5 0 0 15

11 6 0 0 5 3 0 2

## 4 Quellcode

```
let bild = document.getElementById("bild")

let context = bild.getContext("2d")

let große = Math.floor((Math.random() * 26.9) + 4) //Größe des Rätsels

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

var thead = document.createElement("thead")

var headerRow = document.createElement("tr")

var headerCell1 = document.createElement("th")

headerCell1.textContent = große//die erste Spalte sagt aus wie groß das rätsel ist

headerRow.appendChild(headerCell1)

thead.appendChild(headerRow)

table.appendChild(thead)

let zwodliste = []

let teil = []

for (let x = 1; x <= große; x++) { //erstelle leere, Zweidimensionale Liste des Rätsels

    teil = []

    for (let x = 1; x <= große; x++) {

        teil.push("B")

    }

    zwodliste.push(teil)
```

```

}
let parserx = 0
let parsery = 0
let zufall1 = 0
let zufall2 = 0
let zufall3 = 0
let zufall4 = 0
let nicht1 = 0
let nicht2 = 0
let nicht3 = 0
let nicht4 = 0
let überprüfen1 = 0
let überprüfen2 = 0
let überprüfen3 = 0
let überprüfen4 = 0
let y = 1
let beta = true
let alpha = true
let remider = 1
let wiederholung = false
let nichtsplatziert = true
while (beta) { //Wiederhole bis alle Felder eine Zahl haben
    nicht1 = 0
    nicht2 = 0
    nicht3 = 0
    nicht4 = 0
    überprüfen1 = 0
    überprüfen2 = 0
    überprüfen3 = 0
    überprüfen4 = 0
    while (zwodliste[parsery][parserx] != "B") {

```

//setzte den Anfangspunkt des Weges auf eine Zufällige Stelle

```
    parserx = Math.floor(Math.random() * große)
    parsery = Math.floor(Math.random() * große)
}
wiederholung = false
nichtsplatziert = true
for (let x = 1; x <= große; x++) { //wiederhole bis der Weg so Lang ist, wie der Weg Breit
    if (nichtsplatziert) { //Wenn du dich bewegt hast
        if (!wiederholung && x == 1) { //wenn es die erste Stelle des Weges ist
            zwodliste[parsery][parserx] = y //die erste Stelle bekommt die Nummer des Weges
        } else { zwodliste[parsery][parserx] = 0 } //sonst bleibt das Feld leer
        nichtsplatziert = false
    }
    wiederholung = false
    remider = x
    zufall1 = Math.random()
    zufall2 = Math.random()
    zufall3 = Math.random()
    zufall4 = Math.random()
    if (zufall1 <= 0.5) { //wenn 50% Chance
        if (parsery != große - 1) { //wenn die Stelle obendrüber existiert
            if (zwodliste[parsery + 1][parserx] == "B") {
//wenn die stelle obendrüber unbenutzt ist
                parsery = parsery + 1 //ändere die position
                überprüfen1 = 0
                nichtsplatziert = true //es hat sich bewegt
            } else {
                nicht1 = 1 //es geht nicht nach oben
            }
        } else {
            nicht1 = 1 //es geht nicht nach oben
        }
    }
}
```

```

    }
} else if (zufall2 <= 0.5) { //wenn 50% Chance
    if (parserx != große - 1) { //wenn die Stelle rechts Daneben existiert
        if (zwodliste[parsey][parserx + 1] == "B") {
            //wenn die stelle rechts Daneben unbenutzt ist
            parserx = parserx + 1 //ändere die position
            überprüfen2 = 0
            nichtsplaziert = true //es hat sich bewegt
        } else {
            nicht2 = 1 //es geht nicht nach rechts
        }
    } else {
        nicht2 = 1 //es geht nicht nach rechts
    }
} else if (zufall3 <= 0.5) { //wenn 50% Chance
    if (parsey != 0) { //wenn die Stelle untendrunter existiert
        if (zwodliste[parsey - 1][parserx] == "B") {
            //wenn die stelle untendrunter unbenutzt ist
            parsey = parsey - 1 //ändere die position
            überprüfen3 = 0
            nichtsplaziert = true //es hat sich bewegt
        } else {
            nicht3 = 1 //es geht nicht nach unten
        }
    } else {
        nicht3 = 1 //es geht nicht nach unten
    }
}
}
else if (zufall4 <= 0.5) { //wenn 50% Chance
    if (parserx != 0) { //wenn die Stelle links daneben existiert
        if (zwodliste[parsey][parserx - 1] == "B") {

```

//wenn die stelle Links daneben unbenutzt ist

```
        parserx = parserx - 1//ändere die position
        überprüfen4 = 0
        nichtsplatziert = true//es hat sich bewegt
    } else {
        nicht4 = 1//es geht nicht nach links
    }
} else {
    nicht4 = 1//es geht nicht nach links
}
} else { //wenn per zufall keine Richtung getestet wurde
    x = x - 1//der Weg wurde nicht länger
    wiederholung = true
}
if (nicht1 == 1 && überprüfen1 == 0) { //wenn es nach oben nicht geklappt hat
    x = x - 1//der Weg wurde nicht länger
    wiederholung = true
    überprüfen1 = 1
    nicht1 == 0
} else if (nicht2 == 1 && überprüfen2 == 0) { //wenn es nach rechts nicht geklappt hat
    x = x - 1//der Weg wurde nicht länger
    wiederholung = true
    überprüfen2 = 1
    nicht2 == 0
} else if (nicht3 == 1 && überprüfen3 == 0) { //wenn es nach unten nicht geklappt hat
    x = x - 1//der Weg wurde nicht länger
    wiederholung = true
    überprüfen3 = 1
    nicht3 == 0
} else if (nicht4 == 1 && überprüfen4 == 0) { //wenn es nach links nicht geklappt hat
    x = x - 1//der Weg wurde nicht länger
```

```

        wiederholung = true
        überprüfen4 = 1
        nicht4 == 0
    }
    if (überprüfen1 == 1 && überprüfen2 == 1 && überprüfen3 == 1 && überprüfen4 == 1)
    { //wenn es in alle richtungen nicht geklappt hat
        x = große + 1 //der Weg ist beendet
    }
    alpha = true
    for (let s of zwodliste) { //für jedes Feld
        for (let q of s) {
            if (q == "B") { // wenn ein feld unbenutzt ist
                alpha = false //gibt es noch ein Feld, dass es zu füllen gilt
            }
        }
    }
    if (alpha) { //wenn alle Felder eine Zahl bekommen haben
        beta = false
    } //wurden alle Wege gesetzt und es kann mit dem nächsten Schritt weiter Gemacht werden
    }
    zwodliste[parsery][parserx] = y //die Letzte Stelle des Weges bekommt die Nummer des Weges
    y = y + 1 //Wir Schreiten über zum nächsten Weg
}
let quest = y
let toll = 0
let zahl = 0
for (let j = 1; j < y; j++) { //für jeden Weg
    toll = 0
    for (let s = 0; s < zwodliste.length; s++) { //für jedes Feld
        for (let q = 0; q < zwodliste[s].length; q++) {

```



```
        if (j == zwodliste[s][q]) { //Wenn die Nummer des Gesuchten Weges mit der des Feldes  
        übereinstimmt
```

```
            toll = toll + 1 // es gibt ein weiteres ende des Weges
```

```
            if (toll == 1) { //wenn es das erste ende des Weges ist
```

```
                zahl++ //wir haben einen weiteren Weg
```

```
            }
```

```
            zwodliste[s][q] = zahl //ersetzen wir die Nummer des Weges durch die neue
```

```
        }
```

```
    }
```

```
}
```

```
if (toll == 1) { //wenn der Weg nur eine Nummern hat
```

```
    quest = quest - 1 //gibt es insgesamt weniger wege
```

```
    for (let i = 0; i < zwodliste.length; i++) { //für jedes Feld
```

```
        for (let t = 0; t < zwodliste[i].length; t++) {
```

```
            if (zwodliste[i][t] == zahl) { //wenn das Gesuchte wegende gefunden wurde
```

```
                zwodliste[i][t] = 0 //radiere es aus der Menschheitsgeschichte
```

```
            }
```

```
        }
```

```
    }
```

```
    zahl-- //wir haben einen Weg weniger
```

```
}
```

```
}
```

```
var tbody = document.createElement("tbody")
```

```
var row1 = document.createElement("tr")
```

```
tbody.appendChild(row1)
```

```
var cell1 = document.createElement("td")
```

```
cell1.textContent = quest - 1 //die zweite Spalte sagt aus wie viele Zahlpaare es gibt
```

```
row1.appendChild(cell1)
```

```
table.appendChild(tbody)
```

```
for (let x = 1; x <= große; x++) { //für jedes Feld
```

```
    var tbody = document.createElement("tbody")
```

```

var row1 = document.createElement("tr")
tbody.appendChild(row1)
for (let y = 1; y <= große; y++) {
    var cell1 = document.createElement("td")
    cell1.textContent = zwodliste[x - 1][y - 1]//übertrage die Werte aus der Liste in die Tabelle
    row1.appendChild(cell1)
}
table.appendChild(tbody)
}
document.body.appendChild(table)//füge die Tabelle dem Dokument hinzu

```